AFAMRL-TR-80-142

A HUMAN OPERATOR GUNNER MODEL FOR TRACER-DIRECTED ANTIAIRCRAFT ARTILLERY FIRE

KUANG C. WEI
SYSTEMS RESEARCH LABORATORIES, INC.
2800 INDIAN RIPPLE ROAD
DAYTON, OHIO 45440

MARCH 1981

Approved for public release; distribution unlimited.

AIR FORCE AEROSPACE MEDICAL RESEARCH LABORATORY
AEROSPACE MEDICAL DIVISION
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

NOTICES

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Please do not request copies of this report from Air Force Aerospace Medical Research Laboratory. Additional copies may be purchased from:

National Technical Information Service
5285 Port Royal Road
Springfield, Virginia 22161

Federal Government agencies and their contractors registered with Defense Documentation Center should direct requests for copies of this report to:

Defense Documentation Center
Cameron Station
Alexandria, Virginia 22314

TECHNICAL REVIEW AND APPROVAL

AFAMRL-TR-80-142

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

CHARLES BATES, JR.
Chief
Human Engineering Division
Air Force Aerospace Medical Research Laboratory

AIR FORCE/56780/2 April 1981 — 200

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFAMRL-TR-80-142	2. GOVT ACCESSION NO.: AD-A097
3. RECIPIENT'S CATALOG NUMBER: 824	
4. TITLE (and Subtitle): A HUMAN OPERATOR GUNNER MODEL FOR TRACER-DIRECTED ANTIAIRCRAFT ARTILLERY FIRE.	5. TYPE OF REPORT & PERIOD COVERED: Technical Rep[redacted]t
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Kuang C. Wei	8. CONTRACT OR GRANT NUMBER(s): F33615-79-C-0500
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Systems Research Laboratories, Inc. 2800 Indian Ripple Road Dayton, Ohio 45440	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 62202F/6893/04/58
11. CONTROLLING OFFICE NAME AND ADDRESS: Air Force Aerospace Medical Research Laboratory, Aerospace Medical Division, Air Force Systems Command, Wright-Patterson AFB, Ohio 45433	12. REPORT DATE: Mar[redacted]81
	13. NUMBER OF PAGES: 113
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

AAA Tracking
AAA Gunner Model
Tracer Fire
Tracer Tracking
Observer Theory
Pursuit Tracking
Tracker Model
OBS3/ 6
MTQ
Closed-Loop Fire Control

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The design of a mathematical model that simulates the gunner's performance in a manual tracking and firing mode of an antiaircraft artillery (AAA) system is presented. The specific gunnery task modeled involves direct manual rate control of the gun turret with the gunner using an optical sighting system having a line of sight coincident with the gun pointing angle. In this system mode, radar is not used for azimuth, elevation or range data. Rather, the gunner, using visual feedback from the antiaircraft artillery tracer rounds he he has fired, continuo

(continued on reverse)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

340400

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Block 20. Abstract (cont'd)

adjusts weapon pointing in azimuth and elevation to minimize the tracer-to-target error, hence increasing his chances of putting "hits" on the target. The model is based on reduced-order observer theory and consists of a reduced-order observer, a linear feedback controller, and a remnant element. The structure of the model is not only concise, but also describes the key functional roles performed by a gunner in such a system. Short execution time of this model enhances a great deal of its applicability to existing attrition models in evaluation of the survivability of aircraft in tactical engagement scenarios.

A least-squares minimization algorithm is derived to identify the model parameters systematically. It is based on a hill-climbing optimization algorithm and an average approximation method to approximate a system's delayed state variables. The ensembled mean and standard deviation of model predictions for both azimuth and elevation tracking as well as tracer errors are obtained. These data are compared with the human's empirical data obtained from experiments conducted at the Air Force Aerospace Medical Research Laboratory. Results show that model predictions are in close agreement with empirical data for several flyby and maneuvering trajectories. The conclusions are that this gunner model and the parameter identification program can be used accurately and efficiently in the analysis of the effectiveness of AAA weapon systems.

SUMMARY

This report documents the development of a mathematical model which describes the gunner's tracking performance in an AAA tracer-directed manual firing task. Reduced-order observer theory is applied to design this model for the underlying linear time-varying antiaircraft artillery system. The model consists of a reduced-order observer, a linear feedback controller, and a stochastic remnant element. Both the tracking and the tracer errors are considered measurable. The tracer dynamics enter the system in a delayed fashion. The Average Approximation Method is used to solve the closed-loop delay differential equations. A least-squares minimization algorithm is derived to identify the model parameters systematically. Simulation results on model predictions versus empirical data over several input trajectories are included. The gunner model can adequately describe human response in this compensatory tracking and firing task. The gunner model developed here can be used in larger attrition models to evaluate the survivability of aircraft in tactical engagement scenarios.

PREFACE

This report documents a study performed by Systems Research Laboratories, Inc. (SRL), Dayton, Ohio, for the Air Force Aerospace Medical Research Laboratory (AFAMRL), Human Engineering Division, Manned Threat Quantification program. This work was performed under Contract F33615-79-C-0500. The Contract Monitor was Dr. Carroll N. Day, and the Technical Manager was Capt. George J. Valentino. The SRL Project Manager was Mr. Kaile Bishop.

The author extends his appreciation to Dr. Carroll N. Day, Messrs. Walt Summers and Maris M. Vikmanis of the Human Engineering Division of the Air Force Aerospace Medical Research Laboratory, Wright-Patterson AFB, for their valuable comments and discussions.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Section I
INTRODUCTION

The tracking performance of a human operator in an antiaircraft artillery (AAA) system has been studied for several years. Several approaches have been proposed and proven successful in modeling human response in compensatory tracking tasks. Among these are the McRuer Crossover Model (McRuer and Krendel, 1974), Optimal Control Model (Kleinman et al., 1971, 1974), PID Structure Modified Optimal Control Model (Phatak et al., 1976), and Observer Model (Kou et al., 1978). The specific gunnery task modeled involves direct manual rate control of the gun turret with the gunner using an optical sighting system having a line of sight coincident with the gun pointing angle. In this system mode, radar is not used for either azimuth, elevation, or range data. Rather, the gunner, using visual feedback from the antiaircraft artillery tracer rounds he has fired, continuously adjusts weapon pointing in azimuth and elevation to minimize the tracer-to-target error; hence, increases his chances of putting "hits" on the target. Therefore, the subject plays both the role of a tracker and a lead angle computer. This tracking task is greatly complicated by the inclusion of lead angle estimation. An additional measurement channel is available to the gunner to supplement the task through the miss distance of tracer rounds from the target. The tracer dynamics enter the man-machine system in a delayed fashion and complicate the development of a simple but faithful human operator model. In this report, Luenberger observer theory is applied to design an antiaircraft gunner model that consists of a reduced-order observer and a controller. A remnant function that lumps all the random effects from measurement noise and human neuromotor response noise is assumed to be Gaussian with its covariance being a function of target velocity and acceleration. The manned AAA simulation experiment was conducted at the Air Force Aerospace Medical Research Laboratory, Wright-Patterson AFB, Ohio. The empirical tracking and tracer data were used to identify parameters of the gunner model via a least-squares minimization algorithm. The computer simulation of an AAA closed-loop tracking and firing system shows that the gunner model's prediction of tracking error is in close agreement with the empirical data over various input target trajectories.

Section II

AAA TRACKING AND FIRING SYSTEM

In a manual tracking and firing task, the human operator (gunner) perceives both the tracking error and the tracer error on a two-dimensional visual display. The tracking error* e_1 is the difference between the target angle θ_T and the barrel angle θ_B, while the tracer error e_2 is the difference between the target angle θ_T and the projectile ending angle θ_P. For the simulation the projectile flight path ended at the range of the target; each individual tracer round was blanked at this point θ_P and disappeared from the display. This is depicted in Figure 1. The two axes on the display represent the azimuth and elevation components of the tracking and the tracer errors, respectively. The gunner's objective is to align the projectile ending angle θ_P to the target angle θ_T; or to minimize the magnitude of the tracer error to simulate a close hit on the target.

The gunner performs both the tracking and firing task along both the azimuth and elevation axes. Since the structure of the underlying systems for azimuth and elevation axes is similar, they can be decomposed and treated separately. The only difference comes from the azimuth tracking system being nonlinear due to a visual correction factor $\cos(\theta_B)_{EL}$. Figure 2 is the block diagram of an AAA tracking and firing system.

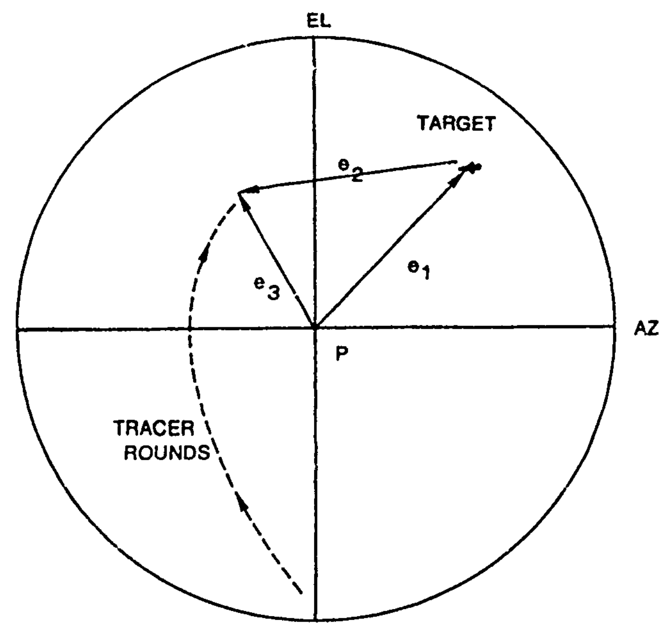
ELEVATION

The elevation barrel dynamics and rate control system can be represented by the transfer function T_{1B}

$$T_{1B}(s) = \frac{\theta_{1B}(s)}{U_1(s)} = \frac{1.34\ (s+16.875)}{s(0.926s^2+s+16.875)} \tag{1}$$

Based on a frequency domain analysis of the input target trajectories (Kou and Glass, 1977), it was found that the frequency bandwidth of all the

*Also referred to as the "lag" angle later in this report.



NOTE: Point "P" is the line of sight of the optical tracking system and is coincident with the gun pointing angle.

Figure 1. Visual Display of Tracking and Tracer Errors

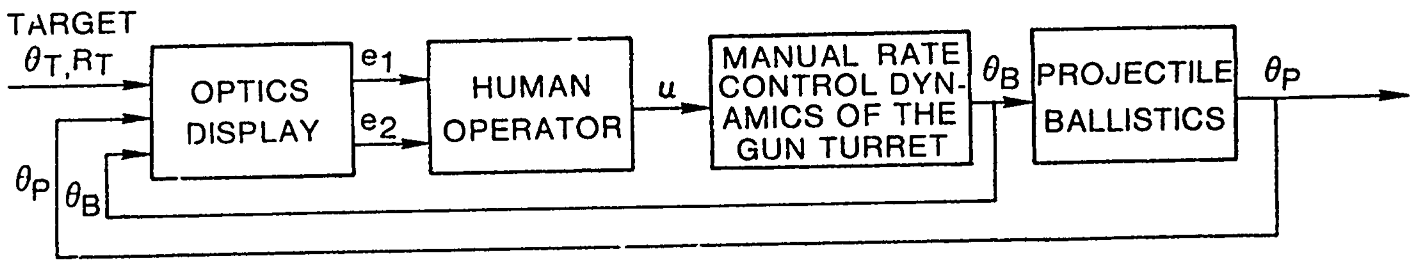


Figure 2. Block Diagram of an AAA Tracking and Firing System

trajectories used in this study is approximately 0.2 Hz. Hence, the above-mentioned transfer function can be approximated by

$$\tilde{T}_{1B}(s) = \frac{1.34}{s} \qquad (2)$$

to simplify the model design and subsequent analysis. The corresponding input-output relation of the elevation barrel system becomes

$$\dot{\theta}_{1B}(t) = 1.34\ u_1(t) \qquad (3)$$

The elevation projectile ballistics can be described by the following equation (see Appendix A):

$$\theta_{1P}(t) = \theta_{1B}(t-\tau) - 0.001\ (5.2\tau+0.486\tau^2)\ \cos\ \theta_{1B}(t-\tau) \qquad (4)$$

where τ is the time-of-flight of the projectile. The second term approximates the elevation drop of a projectile due to gravitation. The τ is determined by the target range $R(t)$ (see Appendix A)

$$\tau \cong \frac{R(t)}{930 - 0.19\ R(t)} \quad \text{for } R(t) \leq 2877 \text{ meters} \qquad (5)$$

and thus is a time-varying quantity.

AZIMUTH

The azimuth barrel dynamics and rate control system can be represented by the following transfer function

$$T_{2B}(s) = \frac{\theta_{2B}(s)}{U_2(s)} = \frac{1.28\ (s+5.312)}{s(0.459s^2+s+5.312)} \qquad (6)$$

Similar to the elevation case, the azimuth transfer function can be approximated by

$$\tilde{T}_{2B}(s) = \frac{1.28}{s} \qquad (7)$$

The corresponding input-output relation of the azimuth barrel plant becomes

$$\dot{\theta}_{2B}(t) = 1.28\, u_2(t) \tag{8}$$

Since there is no gravitational drop in the azimuth axis, the azimuth projectile ballistics can be described simply by

$$\theta_{2P}(t) = \theta_{2B}(t-\tau) \tag{9}$$

The state vector is introduced here

$$\underline{x}_i(t) = \left[x_{i1}(t),\ x_{i2}(t),\ x_{i3}(t)\right]^T,$$

"T" means "the transpose of," with

$$x_{i1}(t) \triangleq \theta_{iT}(t) - \theta_{iB}(t),$$

$$x_{i2}(t) \triangleq \theta_{iT}(t) - \theta_{iP}(t)$$

and

$$x_{i3}(t) \triangleq \dot{\theta}_{iT}(t),$$

where

$$i = 1,\ 2^*$$

*If not otherwise specified, the first subscript index i represents the elevation (i = 1) or azimuth axis (i = 2) while the second index represents the i-th element or row of a matrix.

The following system and measurement equations can be derived from Equations (3)-(4), (8)-(9).

$$\dot{\underline{x}}_i = \underline{A}_i \underline{x}_i + \underline{B}_i\, u_i(t) + \underline{E}_i\, u_i(t-\tau) + \underline{F}_i\, \ddot{\theta}_{iT}(t) + \underline{G}_i(t-\tau) \qquad (10)$$

and

$$\underline{y}_i(t) = \underline{C}_i\, \underline{x}_i(t) \qquad i = 1,2 \qquad (11)$$

where

$$\underline{A}_i = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \qquad \underline{B}_i = \begin{bmatrix} b_i \\ 0 \\ 0 \end{bmatrix} \qquad \underline{E}_i = \begin{bmatrix} 0 \\ e_i(t) \\ 0 \end{bmatrix}$$

$$\underline{F}_i = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad \underline{G}_i = \begin{bmatrix} 0 \\ g_i(t) \\ 0 \end{bmatrix} \qquad \underline{C}_i = \begin{bmatrix} c_i & 0 & 0 \\ 0 & c_i & 0 \end{bmatrix}$$

with

$$b_1 = -1.34$$

$$b_2 = -1.28$$

$$c_1 = 1$$

$$c_2 = \cos\theta_{1B}(t)$$

$$e_1(t) = -1.34 \times (1-\dot{\tau}) \times \left[1 + (0.0052\tau + 0.000486\tau^2)\sin\theta_{1B}(t-\tau)\right]$$

$$e_2(t) = -1.28 \times (1-\dot{\tau})$$

$$g_1(t) = (0.0052 + 0.000972\tau) \times \dot{\tau} \times \cos\theta_{1B}(t-\tau)$$

$$g_2(t) = 0$$

$\ddot{\theta}_{iT}$, u_i, y_{i1}, and y_{i2} denote the elevation or azimuth components of the target acceleration, the gunner's control output and the observed tracking error (lag angle) and tracer error, respectively. By definition, $\theta_{1B}(t)$ can be expressed by $\theta_{1T}(t) - x_{11}(t)$. Thus, the underlying AAA tracking and firing system is not only a time-varying system, but also a nonlinear system with a time-varying delay. In order to have the problem well-posed, several approximations are made to simplify the system equations.

Note that $\theta_{1B}(t)$ is not directly measurable and τ is less than or equal to 7.5 seconds in the simulation experiment, and we may neglect $g_1(t)$ and approximate $e_1(t)$ by

$$\tilde{e}_1(t) = -1.34\ (1-\dot{\tau})\beta_1 \tag{12}$$

where β_1 is a parameter to be identified with other model parameters.

If we introduce a transformation on the states x_{i1} and x_{i2} by $x'_{i1} = c_i x_{i1}$, $x'_{i2} = c_i\ x_{i2}$ then Equations (10)-(11) can be rewritten as follows. Let $\beta_2 = 1$,

$$\dot{\underline{x}}'_i = \underline{A}'_i\ \underline{x}'_i + \underline{B}'_i\ u_i(t) + \underline{E}'_i\ u_i\ (t-\tau) + \underline{F}_i\ \ddot{\theta}_{iT}(t) \tag{13}$$

$$\underline{y}_i(t) = \underline{C}'_i\ \underline{x}'_i(t) \tag{14}$$

where

$$\underline{A}_i' = \begin{bmatrix} \dot{c}_i c_i^{-1} & 0 & c_i \\ 0 & \dot{c}_i c_i^{-1} & c_i \\ 0 & 0 & 0 \end{bmatrix} \qquad \underline{B}_i' = \begin{bmatrix} c_i b_i \\ 0 \\ 0 \end{bmatrix}$$

$$\underline{E}_i' = \begin{bmatrix} 0 \\ c_i b_i (1-\dot{\tau}) \beta_i \\ 0 \end{bmatrix} \qquad \underline{F}_i = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\underline{C}_i' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \qquad i = 1,2$$

Notice that $g_1(t)$ is dropped from Equation (10) and $e_1(t)$ is approximated by the expression shown in Equation (12). The simplified system equations represent a linear time-varying system with a time-varying delay in the control.

Section III

AN AAA OBSERVER GUNNER MODEL

The function of the gunner in an AAA system can be regarded as consisting of two stages to be modeled. In the first stage, the gunner observes continuous signals and makes an estimate of system states (e.g., target velocity) based on his internal model of target motion. In the second stage, the gunner uses the observed and estimated states to form and exercise a proper control action so that a certain objective can be achieved. The former one corresponds to an estimation process, while the latter corresponds to a control process. The reduced-order observer is used in conjunction with a linear state variable feedback (l.s.v.f.) control law to model the gunner's function. The structure of an observer gunner model is shown in Figure 3.

Since the gunner does not have full knowledge of the target dynamics, we may assume that the gunner perceives the target motion (or internal model) as a constant velocity process, i.e., $\ddot{\theta}_T = 0$. Therefore, the equations representing the gunner's internal model of the tracking and firing system can be described by

$$\dot{\underline{x}}_i' = \underline{A}_i'\underline{x}_i' + \underline{B}_i' \, u_i \, (t) + \underline{E}_i' \, u_i \, (t-\tau) \tag{15}$$

$$\underline{y}_i = \underline{C}_i'\underline{x}_i \qquad i = 1,2 \tag{16}$$

Note that $\underline{C}_i'$ is a constant matrix and both x_{i1}, x_{i2} are measurable, the only state that needs to be estimated in order to implement an l.s.v.f. control law is x_{i3}, the target velocity. Reduced-order observer (Luenberger, 1971) is applied to generate an estimate $\hat{x}_{i3}$ of x_{i3} by using Equations (15) and (16). The estimate $\hat{x}_{i3}$ satisfies the following equation:

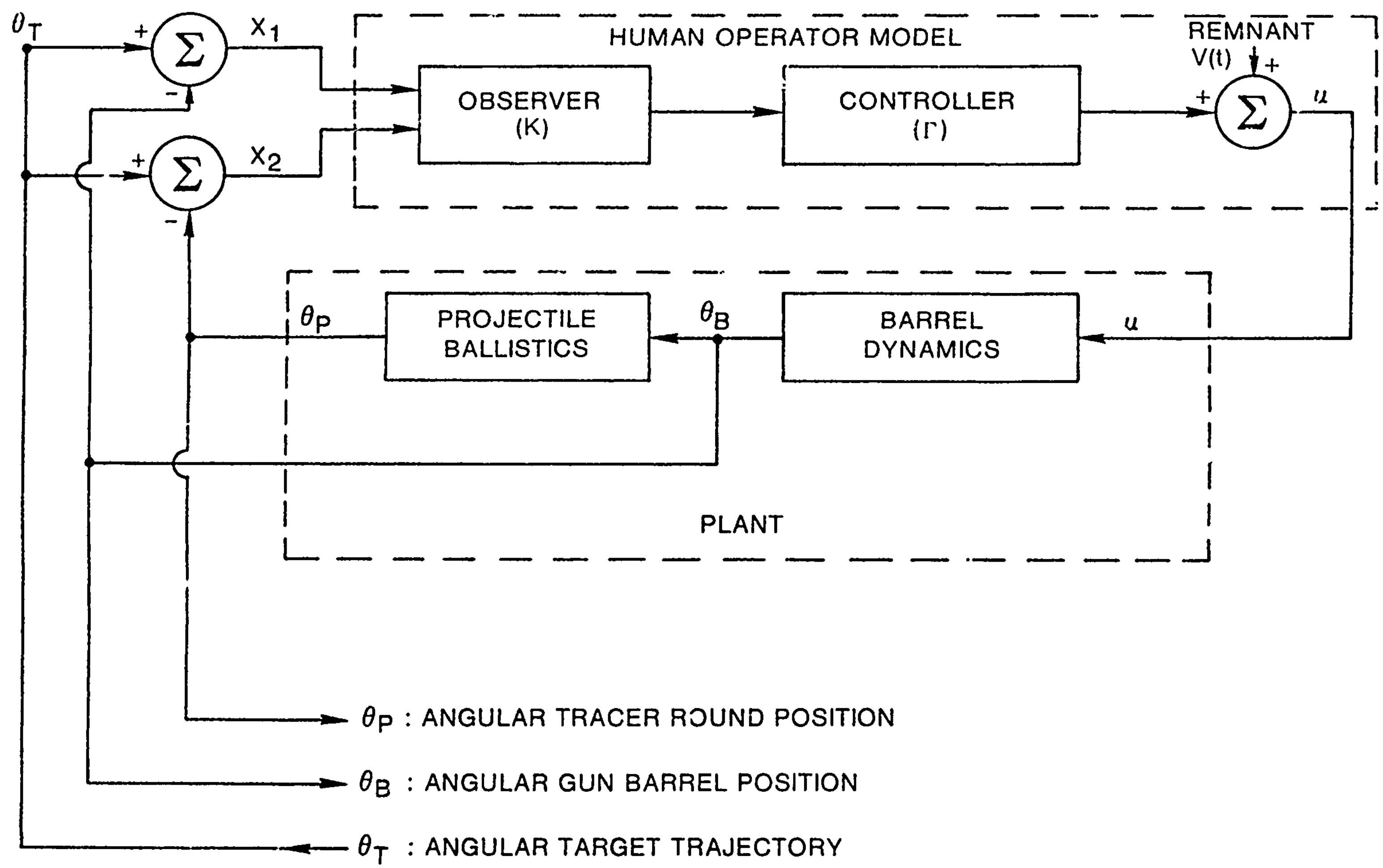


Figure 3. Block Diagram of the Observer Gunner Model

$$\dot{\hat{x}}_{i3} = (k_{i1}+k_{i2})\,\hat{x}_{i3}(t) + k_{i1}\dot{y}_{i1}(t) + k_{i2}\dot{y}_{i2}(t)$$
$$- k_{i1}\dot{c}_i c_i^{-1}\, y_{i1}(t) - k_{i2}\dot{c}_i c_i^{-1} y_{i2}(t)$$
$$- \beta_i\,(1-\dot{\tau})\; k_{i2} b_i\; u_i(t-\tau) - k_{i1} b_i u_i(t) \qquad i = 1,2 \qquad (17)$$

where k_{i1} and k_{i2} are observer gains.

As mentioned earlier in this section, the second function of the gunner is to form a control action based on the observed tracking and tracer error as well as the estimated target velocity. The gunner uses a control manipulator called an "H-grip" to position the gun barrel/optical sight in azimuth and elevation; this rate control system is continuously used by the gunner to minimize the error between the tracer rounds and the target (i.e., minimize x_2 of Figure 3). The objective of the gunner is to minimize the tracer error so that a maximum probability of hit could result. In other words, the gunner's response in the second stage would be to stabilize the underlying system, especially the tracer error $x_{i2}(t)$. A natural control function which may achieve this objective would be a linear feedback control law. Let us consider an l.s.v.f. control of the form

$$u_i(t) = \underline{\Gamma}_i\, \hat{\underline{x}}_i'\,(t) + v_i(t)$$

with

$$\hat{\underline{x}}_i' = \begin{bmatrix} y_{i1} \\ y_{i2} \\ \hat{x}_{i3} \end{bmatrix} \qquad \underline{\Gamma}_i = \begin{bmatrix} \gamma_{i1} \\ \gamma_{i2} \\ \gamma_{i3} \end{bmatrix} \qquad i = 1,2 \qquad (18)$$

where $\underline{\Gamma}_i$ is a vector of controller gains to be identified, $\hat{\underline{x}}_i'$ is a vector of measurable states and estimated state.

A remnant function $v_i(t)$ is introduced here to take into account the random effects due to the measurement noise, the neuromotor noise, target uncertainty, modeling error, difficulty of tracking, etc. It is modeled as a white noise with zero mean and with covariance function

$$E\left[v_i(t)\, v_i(s)\right] = \left[\alpha_{i1} + \alpha_{i2}\left|\hat{\dot{\theta}}_{iT}(t)\right| + \alpha_{i3}\left|\hat{\ddot{\theta}}_{iT}(t)\right|\right] \delta(t-s)$$

$$i = 1,2 \tag{19}$$

for all t and s. α_{ij} are nonnegative model parameters to be determined. $\hat{\dot{\theta}}_{iT}$ and $\hat{\ddot{\theta}}_{iT}$ are estimated target angle rate and acceleration, respectively.

The model Equations (17) and (18) describe the gunner's response in the estimation and the control stage, respectively. The overall equations of the man-in-the-loop AAA tracking and firing system are obtained by combining Equations (17) and (18) with Equations (13) and (14) of the actual tracking and firing system. If we define a new state vector

$$\underline{X}_i = \left[y_{i1},\ y_{i2},\ x_{i3},\ x_{i3} - \hat{x}_{i3}\right]^T \tag{20}$$

then the state equation of the closed-loop system becomes

$$\dot{\underline{X}}_i = \underline{A}_i\, \underline{X}_i(t) + \underline{D}_i\, \underline{X}_i(t-\tau) + \underline{F}_i \ddot{\theta}_{iT} + \underline{B}_i v_i(t) + \underline{E}_i v_i(t-\tau) \tag{21}$$

where

$$\underline{A}_i = \begin{bmatrix} \dot{c}_i c_i^{-1} + b_i c_i \gamma_{i1} & b_i c_i \gamma_{i2} & (1+b_i\gamma_{i3})c_i & -b_i c_i \gamma_{i3} \\ 0 & \dot{c}_i c_i^{-1} & c_i & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -k_i c_i \end{bmatrix}$$

$$\underline{D}_i = b_i c_i \beta_i \ (1-\dot{\tau}) \cdot \begin{bmatrix} 0 & 0 & 0 & 0 \\ \gamma_{i1} & \gamma_{i2} & \gamma_{i3} & -\gamma_{i3} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\underline{F}_i = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \qquad \underline{B}_i = \begin{bmatrix} b_i c_i \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \underline{E}_i = \begin{bmatrix} 0 \\ b_i c_i \ (1-\dot{\tau})\beta_i \\ 0 \\ 0 \end{bmatrix}$$

$$k_i \triangleq k_{i1} + k_{i2} \qquad\qquad i = 1,\ 2$$

There are eight* model parameters in total, i.e., β_i, k_i, γ_{i1}, γ_{i2}, γ_{i3} α_{i1}, α_{i2}, and α_{i3} that need to be determined in the gunner model.

---

*For the azimuth, only seven parameters need to be identified since $\beta_2 = 1$ by definition.

Section IV

A LEAST-SQUARES IDENTIFICATION PROGRAM

Since the last two components of $\underline{X}_i$ are independent from the first two components as indicated in Equation (21), the system equations can be decoupled into the following form:

$$\begin{cases} \dot{x}_{i3}(t) = \ddot{\theta}_{iT}(t) \\ \dot{x}_{i4}(t) = -k_i c_i \, x_{i4}(t) + \ddot{\theta}_{iT}(t) \end{cases} \tag{22}$$

$$\begin{cases} \dot{x}_{i1}(t) = (\dot{c}_i c_i^{-1} + b_i c_i \gamma_{i1}) \, x_{i1}(t) + b_i c_i \, \gamma_{i2} x_{i2}(t) \\ \qquad + b_i c_i v_i(t) + f_{i1}(t) \\ \dot{x}_{i2}(t) = \dot{c}_i c_i^{-1} \, x_{i2}(t) + b_i c_i \beta_i (1-\dot{\tau}) \left[\gamma_{i1} x_{i1}(t-\tau) \right. \\ \qquad \left. + \gamma_{i2} x_{i2}(t-\tau) + v_i(t-\tau)\right] + f_{i2}(t) \end{cases} \tag{23}$$

with

$$f_{i1}(t) = (1+b_i\gamma_{i3}) \, c_i \, x_{i3}(t) - b_i c_i \gamma_{i3} x_{i4}(t)$$

$$f_{i2}(t) = c_i x_{i3}(t) + b_i c_i \beta_i \, (1-\dot{\tau}) \, \gamma_{i3}\left[x_{i3}(t-\tau) - x_{i4}(t-\tau)\right] \tag{24}$$

$$i = 1,2$$

Note that Equation (23) represents a linear-time varying system with time-varying delay. Since an explicit solution for this type of delay differential equation is usually not available, the Average Approximation Method is applied to approximate Equation (23) by the following ordinary differential equation (Banks and Burns, 1978).

$$\dot{\underline{W}}_i^N(t) = \underline{A}_i^N(t) \, \underline{W}_i^N(t) + \underline{M}_i^N \, \underline{g}_i(t) \tag{25}$$

where

$$
\underline{A}_i^N = \left[\begin{array}{cc:cc:c:c:c:c}
\dot{c}_i c_i^{-1} + b_i c_i \gamma_{i1} & b_i c_i \gamma_{i2} & 0 & 0 & 0. & ..0 & 0 & 0 \\
0 & \dot{c}_i c_i^{-1} & 0 & 0 & 0. & ..0 & b_i c_i \beta_i (1-\dot{\tau}) \gamma_{i1} & b_i c_i \beta_i (1-\dot{\tau}) \gamma_{i2} \\
\hdashline
\frac{N}{\tau} & 0 & -\frac{N}{\tau} & 0 & 0. & .. & 0 & 0 \\
0 & \frac{N}{\tau} & 0 & -\frac{N}{\tau} & 0. & .. & 0 & 0 \\
\hdashline
\multicolumn{2}{c:}{0} & & & \multicolumn{2}{c:}{\ddots} & 0 & 0 \\
\hdashline
 & & & & & \frac{N}{\tau} \quad 0 & -\frac{N}{\tau} & 0 \\
\multicolumn{2}{c:}{0} & 0 & 0 & 0 & 0 \quad \frac{N}{\tau} & 0 & -\frac{N}{\tau}
\end{array}\right]
\begin{array}{l} \\ \\ \\ \left.\begin{array}{l} \\ \\ \\ \end{array}\right\} 2(N+1) \\ \\ \\ \end{array}
$$

$$\underbrace{\qquad\qquad\qquad}_{2\ (N+1)}$$

$$\underline{M}_i^N = \overbrace{\begin{bmatrix} 1 & 0 & 0 & \cdots\cdots & 0 \\ 0 & 1 & 0 & \cdots\cdots & 0 \end{bmatrix}^T}^{2\,(N+1)} \qquad i = 1,2$$

$$\underline{g}_i = \begin{bmatrix} f_{i1}(t) + b_i c_i v_i(t) \\ f_{i2}(t) + b_i c_i \beta_i (1-\dot{\tau})\; v_i(t-\tau) \end{bmatrix}$$

and

$$\underline{W}_i^N = \Big[x_{i1}(t),\ x_{i2}(t),\ x_{i1}\left(t - \frac{N-1}{N}\tau\right),\ x_{i2}\left(t - \frac{N-1}{N}\tau\right),\ \ldots$$

$$x_{i1}(t-\tau),\ x_{i2}(t-\tau) \Big]^T$$

is a $2(N+1) \times 1$ vector consisted of the original states $x_{i1}(t)$, $x_{i2}(t)$ and the approximated delayed states of order N.

Based on a comparison study, the time history of $x_{i1}(t)$ and $x_{i2}(t)$ fell within 5 percent of each other when N = 1 was used versus N = 10. Therefore N = 1 was adopted to save computer memory and reduce computation time.* For N = 1, we have

$$\dot{\underline{W}}_i^1(t) = \underline{A}_i^1(t)\ \underline{W}_i^1(t) + \underline{M}_i^1\ \underline{g}_i(t) \tag{26}$$

*Memory saving is up to 50 percent and computation time can be reduced up to 75 percent.

$$\underline{A}_i^1 = \begin{bmatrix} \dot{c}_i c_i^{-1} + b_i c_i \gamma_{i1} & b_i c_i \gamma_{i2} & 0 & 0 \\ 0 & \dot{c}_i c_i^{-1} & b_i c_i \gamma_{i1} \beta_i (1-\dot{\tau}) & b_i c_i \gamma_{i2} \beta_i (1-\dot{\tau}) \\ \frac{1}{\tau} & 0 & -\frac{1}{\tau} & 0 \\ 0 & \frac{1}{\tau} & 0 & -\frac{1}{\tau} \end{bmatrix}$$

$$\underline{M}_i^1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

In order to identify model parameters, we first take the expectation of Equation (26) to obtain the equation governing the mean of states

$$\dot{\overline{\underline{W}}}_i^1 (t) = \underline{A}_i^1(t)\, \overline{\underline{W}}_i(t) + \underline{M}_i^1\, \underline{f}_i(t) \tag{27}$$

where

$$\overline{\underline{W}}_i^1(t) = \left\{ E\left[x_{i1}(t)\right],\ E\left[x_{i2}(t)\right],\ E\left[x_{i1}(t-\tau)\right],\ E\left[x_{i2}(t-\tau)\right] \right\}^T$$

and

$$\underline{f}_i(t) = \left[f_{i1}(t),\ f_{i2}(t)\right]^T$$

Notice that the remnant terms $v_i(t)$ and $v_i(t-\tau)$ disappear because of the assumption of its zero mean. The first and second component of $\overline{\underline{W}}_i^1$, w_{i1} and w_{i2}, are the model prediction of ensembled mean of tracking and tracer error, respectively.

On the other hand, the covariance matrix $\underline{P}_i(t)$ satisfies the following equation:

$$\dot{\underline{P}}_i(t) = \underline{A}_i^1(t)\ \underline{P}_i(t) + \underline{P}_i(t)\ \underline{A}_i^{1T}(t) + \underline{D}_i(t)\ \underline{Q}_i(t)\ \underline{D}_i^T(t) \qquad (28)$$

where

$$\underline{P}_i(t) = E\left\{\left[\underline{W}_i^1(t) - \overline{\underline{W}}_i^1(t)\right]\ \left[\underline{W}_i^1(t) - \overline{\underline{W}}_i^1(t)\right]^T\right\}$$

$$\underline{D}_i(t) = \begin{bmatrix} b_i c_i & 0 \\ 0 & b_i c_i \beta_i (1-\tau) \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\underline{Q}_i(t) = \begin{bmatrix} \alpha_{i1} + \alpha_{i2}\left|\hat{\dot{\theta}}_{iT}(t)\right| + \alpha_{i3}\left|\hat{\ddot{\theta}}_{iT}(t)\right| & 0 \\ 0 & \alpha_{i1} + \alpha_{i2}\left|\hat{\dot{\theta}}_{iT}(t-\tau)\right| + \alpha_{i3}\left|\hat{\ddot{\theta}}_{iT}(t-\tau)\right| \end{bmatrix}$$

The first diagonal element p_{i11} of $\underline{P}_i(t)$ is the square of the ensembled standard deviation of tracking errors. Similarly, the second diagonal element p_{i22} of $\underline{P}_i(t)$ is the square of the ensembled standard deviation of tracer errors. By solving the matrix differential Equation (28), and defining

$$s_{i1}(t) = p_{i11}(t)^{\frac{1}{2}},\ s_{i2}(t) = p_{i22}(t)^{\frac{1}{2}} \qquad (29)$$

the model prediction of standard deviation of tracking and tracer errors is obtained.

The parameters of the gunner model are determined via a least-squares curve-fitting identification program. The reference curves to be fitted in

the curve-fitting program are obtained from empirical tracking and tracer data collected in the manned simulation experiments. These experiments were conducted on an AAA simulator at the Air Force Aerospace Medical Research Laboratory. Several simulated nonmaneuvering (i.e., flyby) and maneuvering (i.e., recon) aircraft trajectories were used as target trajectories for these experiments. Let $\bar{x}_{i1}(t)$ and $\bar{x}_{i2}(t)$ be the empirical means of tracking and tracer errors, $\bar{s}_{i1}(t)$ and $\bar{s}_{i2}(t)$ be the corresponding standard deviations. These empirical means and standard deviations were obtained by averaging and computing the variance of the empirical data of forty experimental simulation runs with the same target trajectory and the same subject.

The parameters were identified by minimizing the cost function $J(k, \underline{\Gamma}, \underline{\alpha}, \beta,)$ defined as follows:

$$\min_{k,\beta,\underline{\Gamma},\underline{\alpha}} J_i(k, \beta, \underline{\Gamma}, \underline{\alpha}) = \min_{k,\beta,\underline{\Gamma},\underline{\alpha}} \sum_{j=1}^{2} \int_{t_o}^{t_f} \left\{ \left[c_i^{-1} w_{ij}(t) - \bar{x}_{ij}(t) \right]^2 + \ell_i \left[c_i^{-1} s_{ij}(t) - \bar{s}_{ij}(t) \right]^2 \right\} dt \qquad i = 1,2 \tag{30}$$

where t_o is the time when the first tracer round reaches the range of the target, t_f is the time when the last tracer round is fired. The parameters to be determined are k, β, $\underline{\Gamma}$ and $\underline{\alpha}$ as defined in Equations (13), (17)-(19). ℓ_i is a positive weighting factor chosen to be one in the identification runs.

The cost function J_i consists of four contributing terms. These terms are, respectively:

- The integral of the square errors between the empirical mean tracking error $\bar{x}_{i1}$ and the model prediction $c_i^{-1} w_{i1}$.

- The square errors between the empirical mean tracer error $\bar{x}_{i2}$ and the model prediction $c_i^{-1}w_{i2}$.

- The square errors between the empirical standard deviation $\bar{s}_{i1}$ of tracking error and the model prediction $c_i^{-1}s_{i1}$.

- Finally, the square errors between the empirical standard deviation $\bar{s}_{i2}$ of tracer error and the model prediction $c_i^{-1}s_{i2}$.

Therefore, the model parameters will be identified such that model prediction of tracking and tracer error and their standard deviations will fit with the empirical counterparts simultaneously. This is a fairly complex minimization problem which would be too cumbersome and inefficient to use the Gauss-Newton gradient method. A direct search method based on Rosenbrock's hill-climbing minimization algorithm (Rosenbrock, 1966) was derived (see Appendix B). This method adjusts parameter values along a set of orthonormal axes iteratively to search for the least cost J_i, before a rotation of axes is performed and a new series of searches are launched. The iterative process will continue until the changes in parameter values are smaller than a prespecified value. A computer curve-fitting program was written which implements the previously mentioned minimization algorithm. Some merits of the identification program are that it has simple computation requirements, provides a systematic search, and gives reasonably fast convergence speed.

A set of converged parameter values were obtained by this identification program for both the elevation and azimuth gunner model. These values are listed in Table 1.

The preceding parameter values were obtained by applying the empirical data from the Recon flight path. Notice that the gunner model depends on the dynamics of the barrel plant. For different AAA tracking systems, the identification program can be used to determine parameter values for the corresponding gunner model.

TABLE 1. PARAMETERS FOR ELEVATION AND AZIMUTH GUNNER MODEL

Parameter / Gunner Model for	Observer Gain	Projectile Slope	Controller Gains			Coefficients of Remnant Covariance Function		
	k	β	γ_1	γ_2	γ_3	α_1	α_2	α_3
Elevation	0.79531	1.2862	1.4143	1.4932	0.90605	0.1E-4	0.2207E-5	0.76807E-3
Azimuth	557.15	1.	0.16447	0.61506	0.93641	0.15634E-4	0.64182E-4	0.005289

31

Section V
COMPUTER SIMULATION RESULTS

The gunner model described in the last section was implemented on a CDC CYBER 175 computer to simulate the man-in-the-loop AAA tracking and firing task. The parameters of the gunner model listed in Table 1 were used in a simulation of gunner's response for five typical operational trajectories, including both flyby as well as maneuvering trajectories. In order to avoid using convolution integration (to reduce computation time) in solving Equations (27) and (28), we discretize the system equations into (for simplicity of notation we may drop the subscript i and the superscript 1 from here on),

$$\underline{W}_{n+1} = \underline{\Phi}_n \underline{W}_n + \underline{H}_n \underline{f}_n + \underline{L}_n \underline{v}_n \qquad (31)$$

where

$$\underline{W}_{n+1} = \underline{W}\left(t_{n+1}\right) \text{ with } t_{n+1} = t_o + (n+1)\Delta$$

and $\Delta = 0.06$ seconds.

$$\underline{\Phi}_n = \exp\left[\underline{A}(t_n)\ \Delta\right],$$

$$\underline{H}_n = \int_o^{\Delta} \exp\left[\underline{A}(t_n) \cdot \sigma\right] d\sigma$$

$$\underline{L}_n = \int_o^{\Delta} \exp\left[\underline{A}(t_n) \cdot \sigma\right] d\sigma \cdot \underline{D}(t_n)$$

$$\underline{v}_n = \left[v(t_n),\ v(t_n - \tau)\right]^T$$

and $\underline{f}_n = \underline{f}(t_n)$ as defined in Equation (24). Notice that $\underline{v}_n$ defined here is a random sequence with the following properties.

$$E\left[\underline{v}_n\right] = 0, \; E\left[(\underline{v}_n)\;(\underline{v}_n)^T\right] = \frac{1}{\Delta}\,\underline{Q}(t_n) \tag{32}$$

Taking the expectation of both sides of Equation (31) we obtain

$$\overline{\underline{W}}_{n+1} = \underline{\Phi}_n\,\overline{\underline{W}}_n + \underline{H}_n\,\underline{f}_n \tag{33}$$

Denote $\underline{P}_{n+1}$ as the covariance matrix of $\underline{W}_{n+1}$, then the solution of Equation (28) can be expressed in the following discrete form:

$$\underline{P}_{n+1} = \underline{\Phi}_n\,\underline{P}_n\,\underline{\Phi}_n^{\;T} + \frac{1}{\Delta}\,\underline{L}_n\,\underline{Q}_n\,\underline{L}_n^{\;T} \tag{34}$$

The first and second elements of $\overline{\underline{W}}_{n+1}$ in Equation (33) are the model prediction of the mean tracking and tracer error correspondingly. The first and second diagonal elements of $\underline{P}_{n+1}$ in Equation (34) are the model prediction of the variance of tracking and tracer errors.

A computer program was developed which uses the recursive Equations (33) and (34) to simulate a closed-loop AAA tracking and firing task. Inputs to the simulation program are the time history of range and acceleration of the target aircraft, plus the initial angular position and velocity of the target. Outputs of the simulation program are model predicted mean tracking error and its standard deviation. Five trajectories including flyby and maneuvering trajectories were used in this study and are shown in Figure 4a-4e. The AAA weapon system was located at the origin of the x-y plane, with aircraft altitude measured along the -z axis. The increment of each of the three axes is 1000 feet. A detailed description of the characteristics of these trajectories can be found in (Rolek, 1977).

Simulation results are shown in Figures 5 through 24 for the trajectories listed in Figure 4. The solid curve in these figures is the empirical data which are obtained by averaging the results of 40 experimental simulation runs. The dashed curve* is the model prediction of ensembled mean and

*Initial conditions used in these simulation runs are the same as the empirical data. In general, initial conditions are computed by empirical formulas referred in Appendix A.

standard deviation. Figures 5 through 8 show the results of elevation mean and standard deviation, azimuth mean and standard deviation of both tracking errors and tracer errors for the Recon trajectory. These figures show that the designed gunner model can provide consistent prediction of the human's empirical tracking data as well as the tracer error data which result when tracking error propagates through the barrel plant. The remaining figures show that similar agreement between predicted and empirical data holds for the other four trajectories, too. The author concluded that for a given AAA weapon system, the same set of parameter values can be used in the gunner model to predict the human operator's tracking and tracer errors for all simulated target trajectories. In other words, the gunner model designed here is a predictive model.

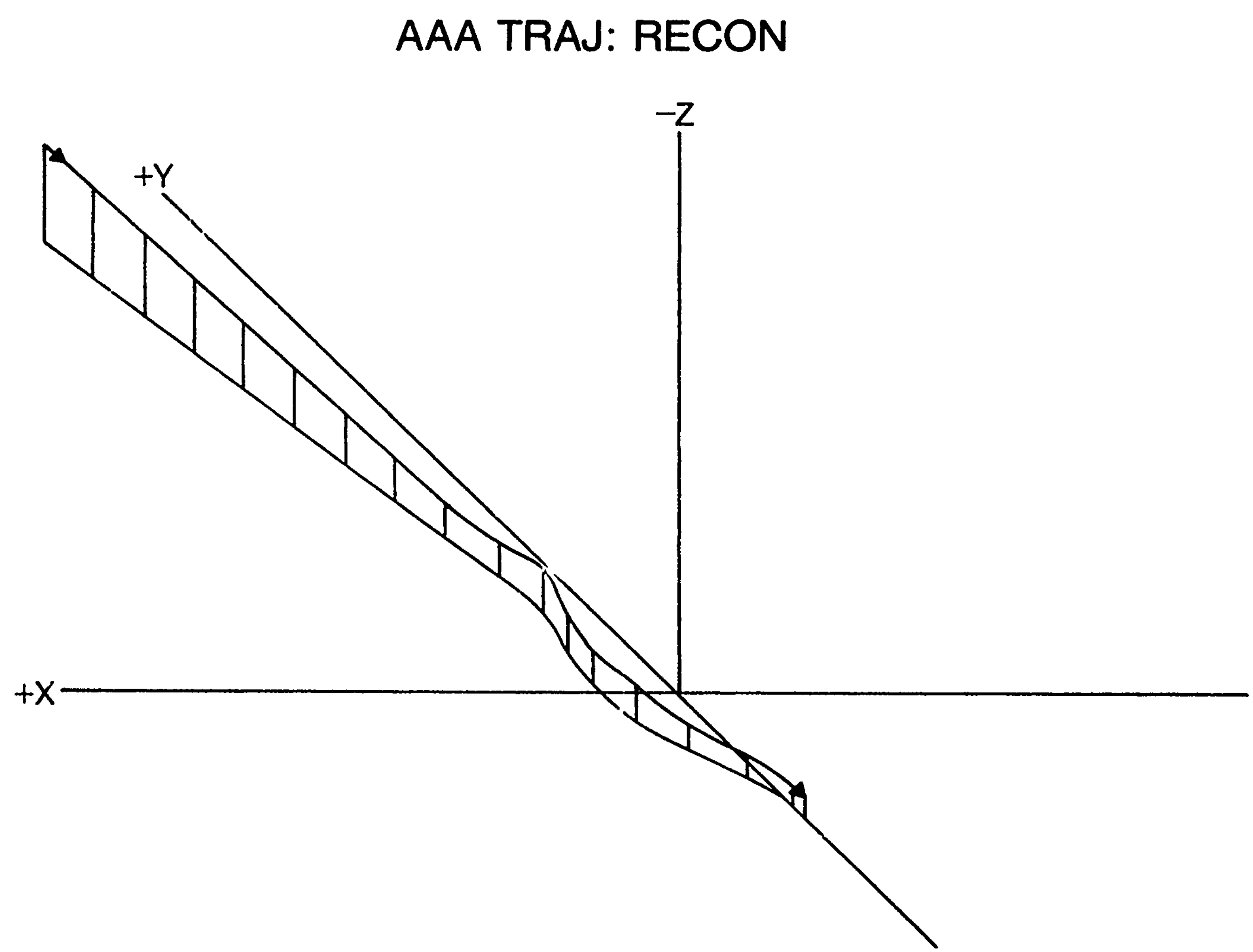


Figure 4a. Flyby and Maneuvering Target Trajectories

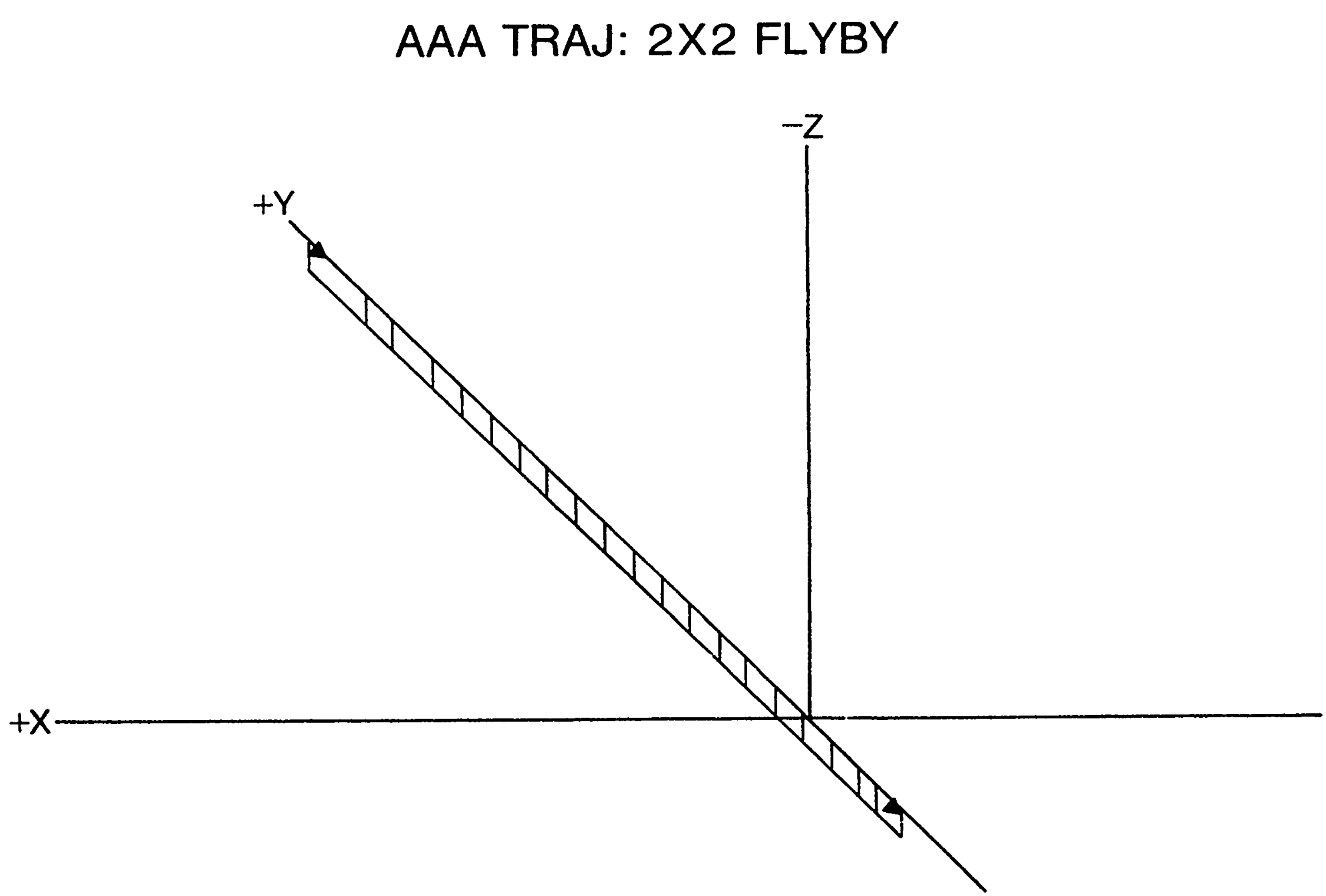


Figure 4b. Flyby and Maneuvering Target Trajectories

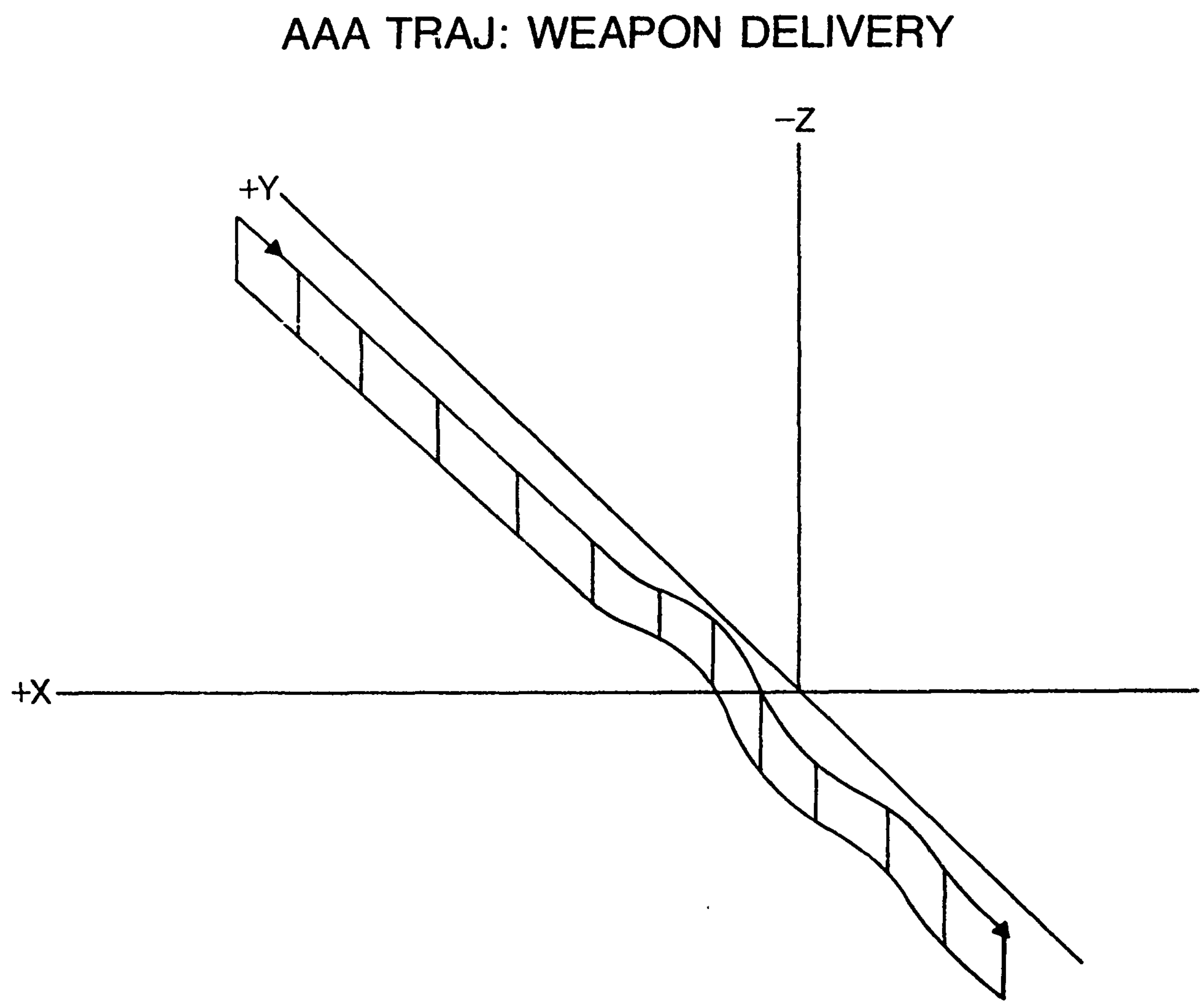


Figure 4c. Flyby and Maneuvering Target Trajectories

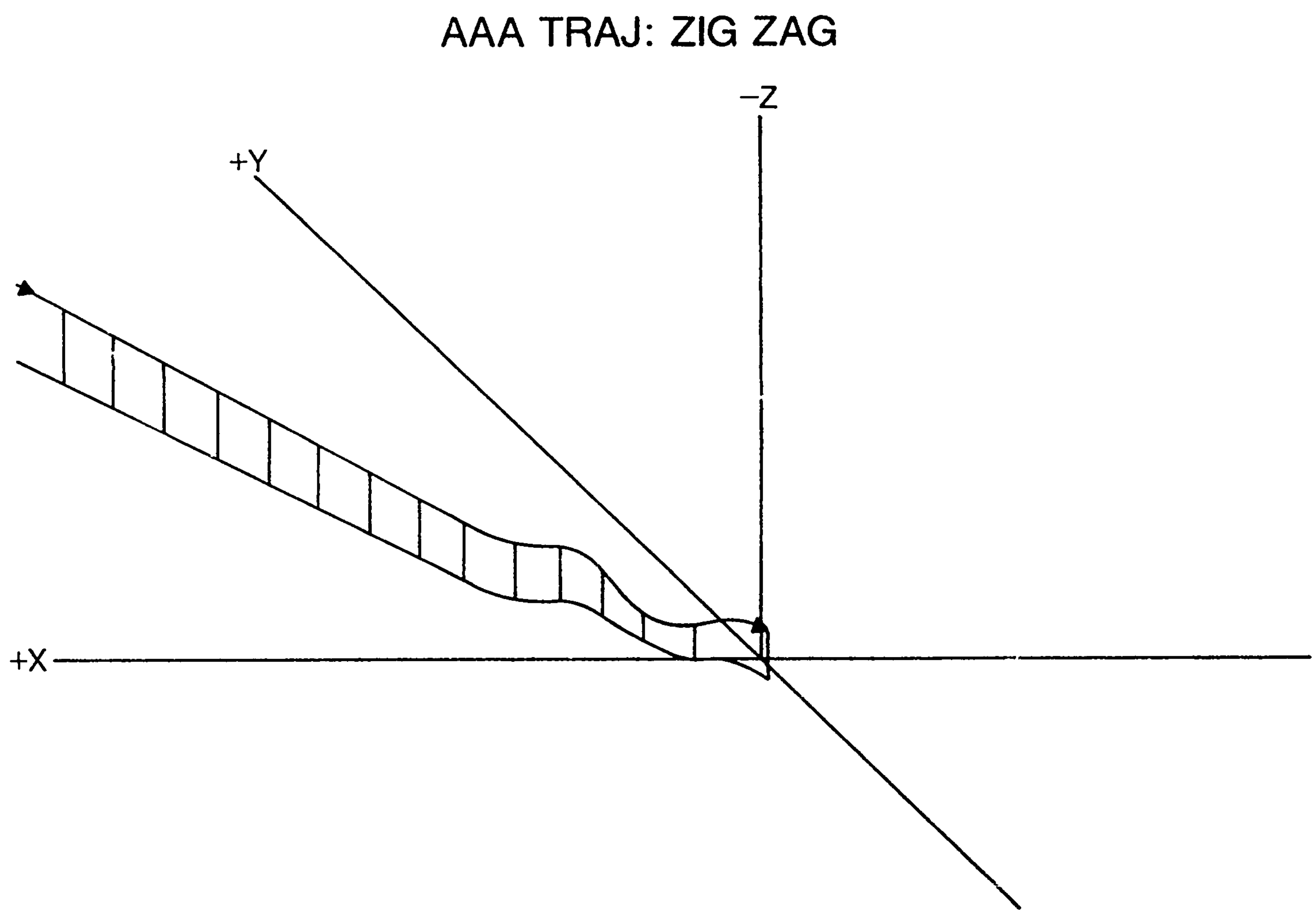


Figure 4d. Flyby and Maneuvering Target Trajectories

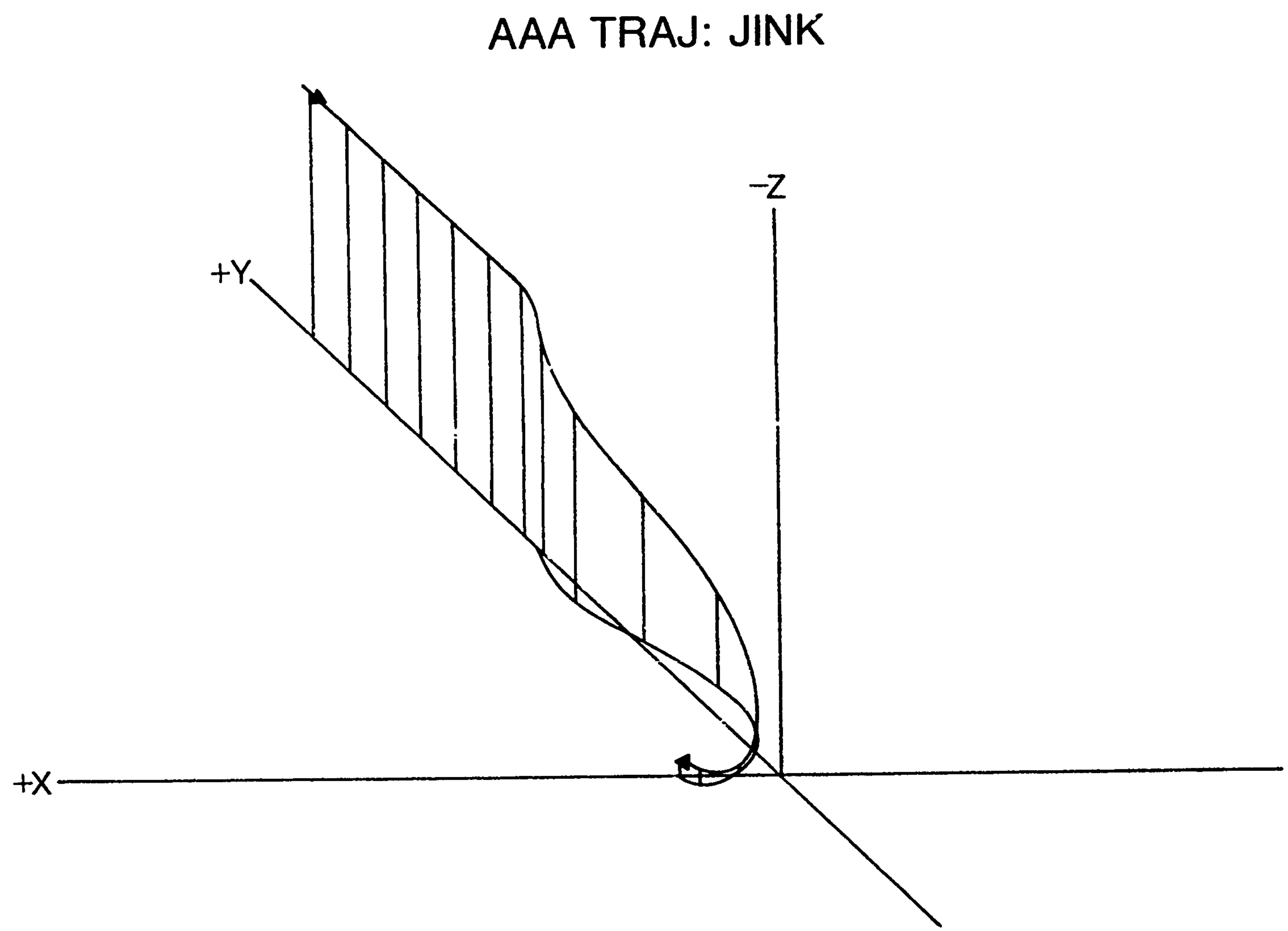


Figure 4e. Flyby and Maneuvering Target Trajectories

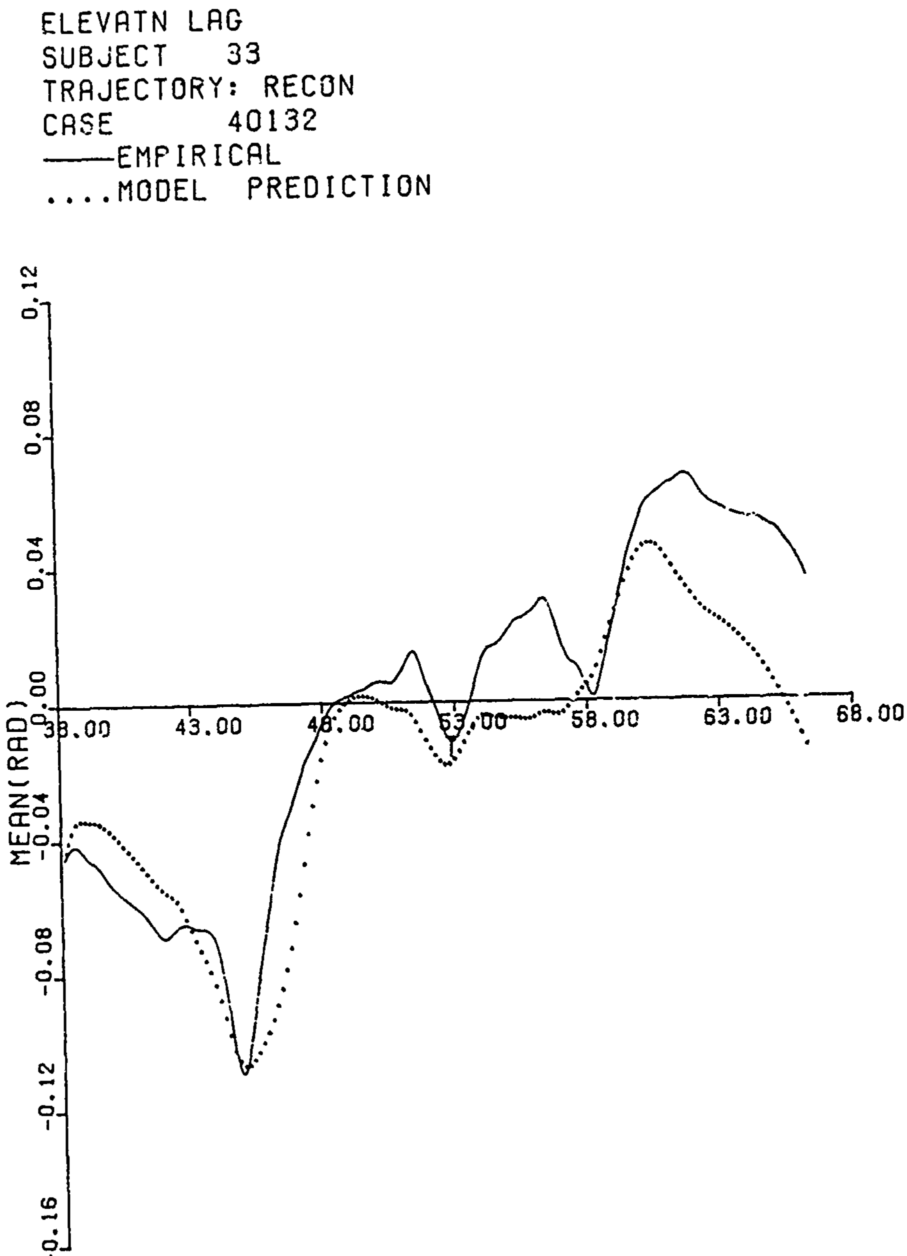


Figure 5a. Mean Tracking Error--Elevation--Recon

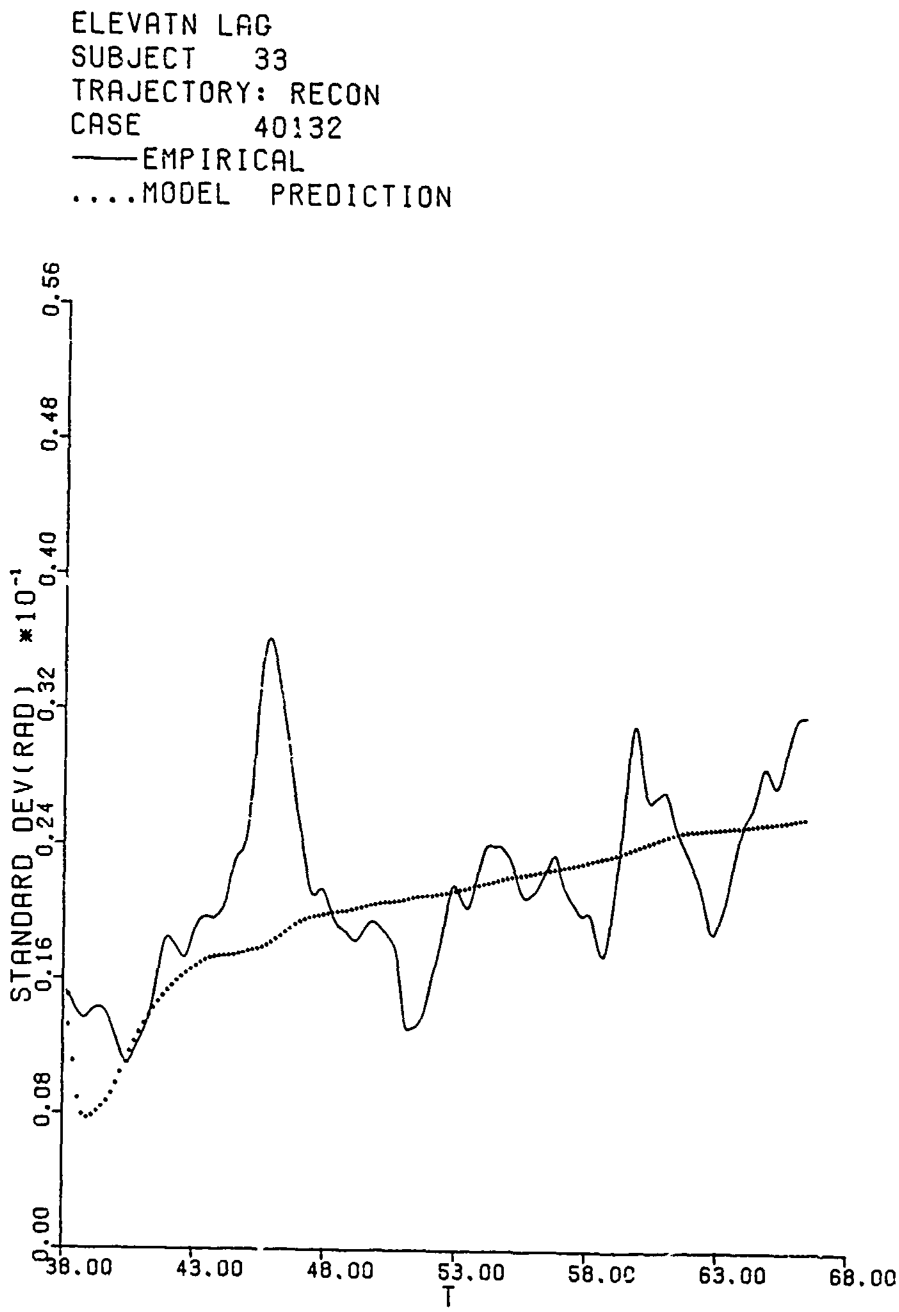


Figure 5b. Standard Deviation of Tracking Error--Elevation--Recon

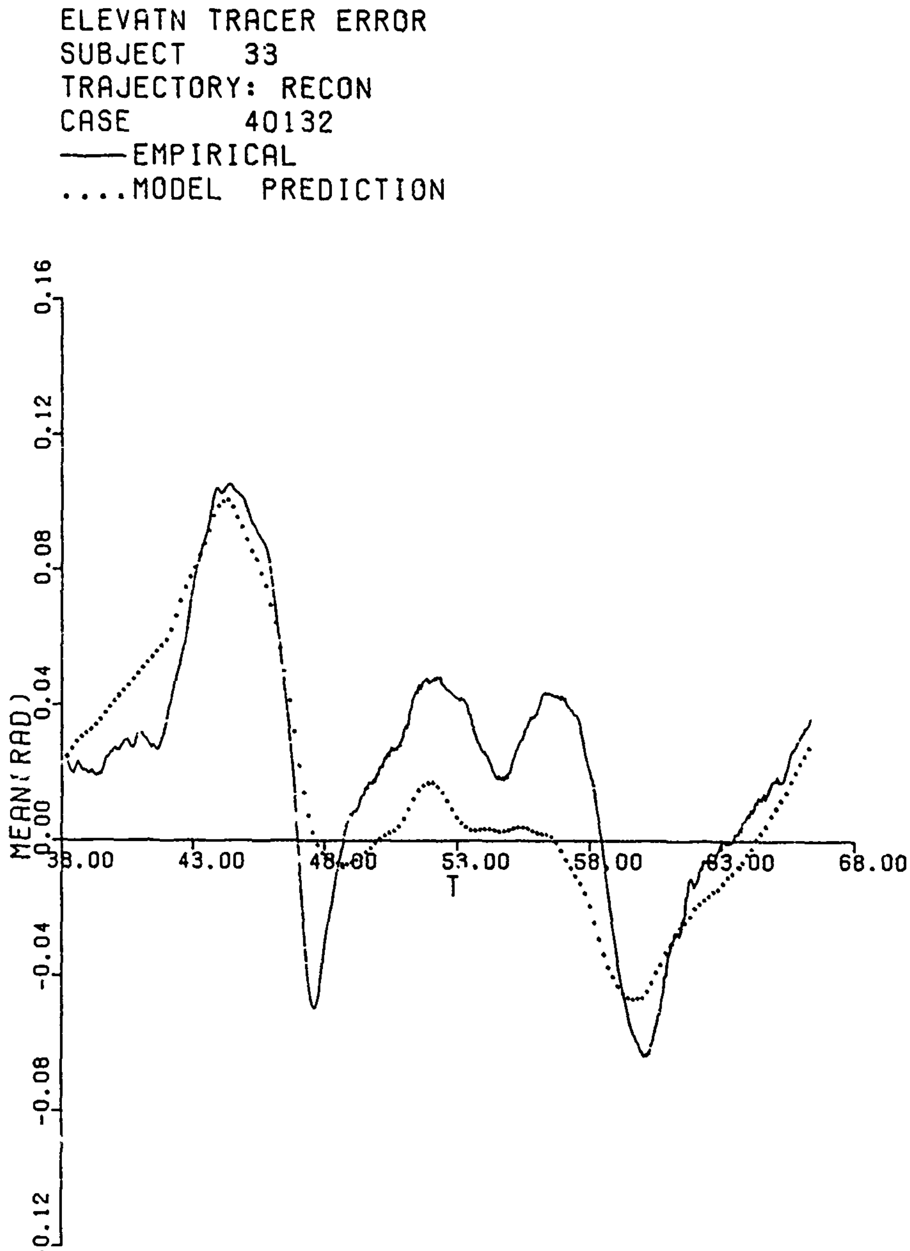


Figure 6a. Mean Tracer Error--Elevation--Recon

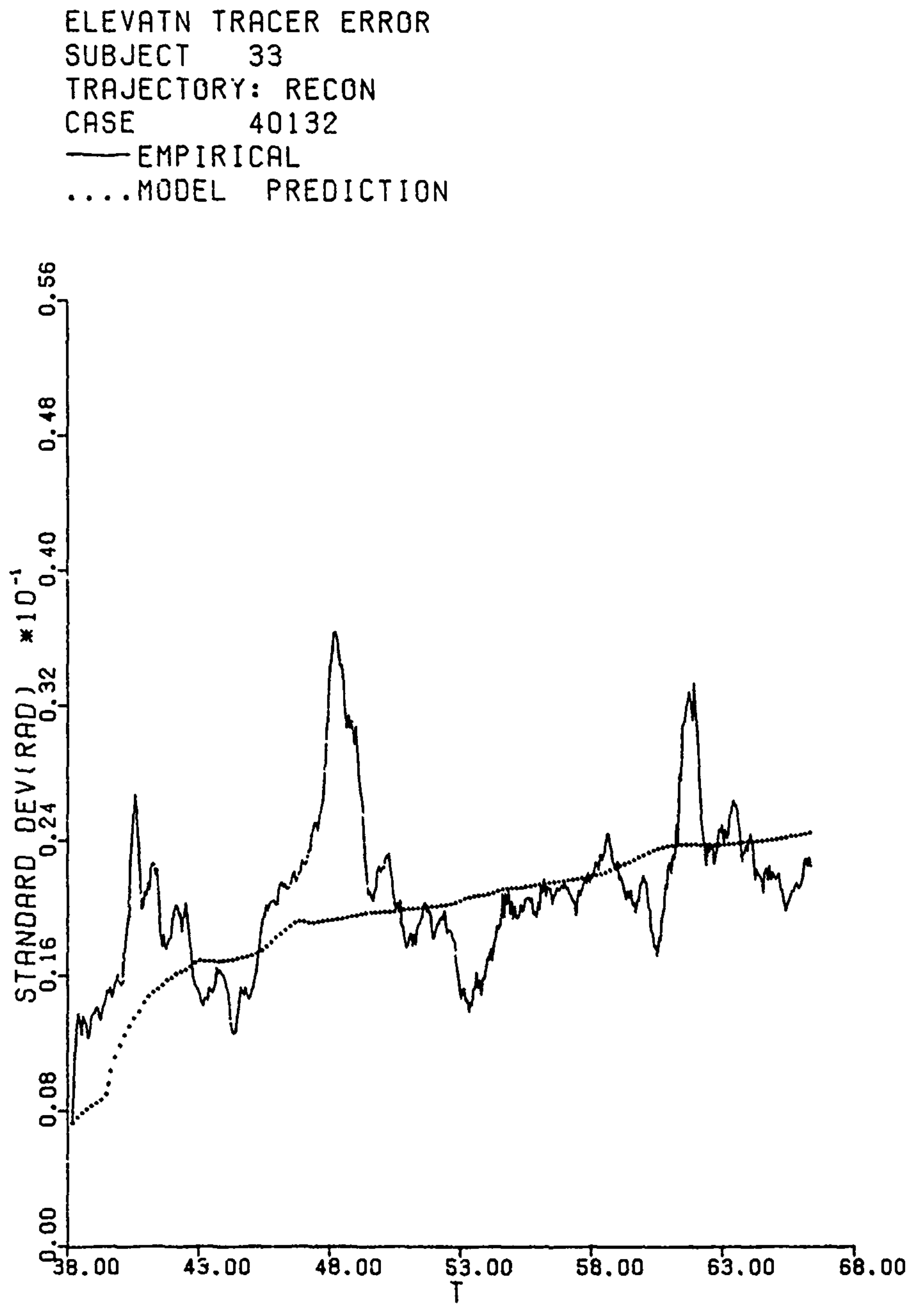


Figure 6b. Standard Deviation of Tracer Error--Elevation--Recon

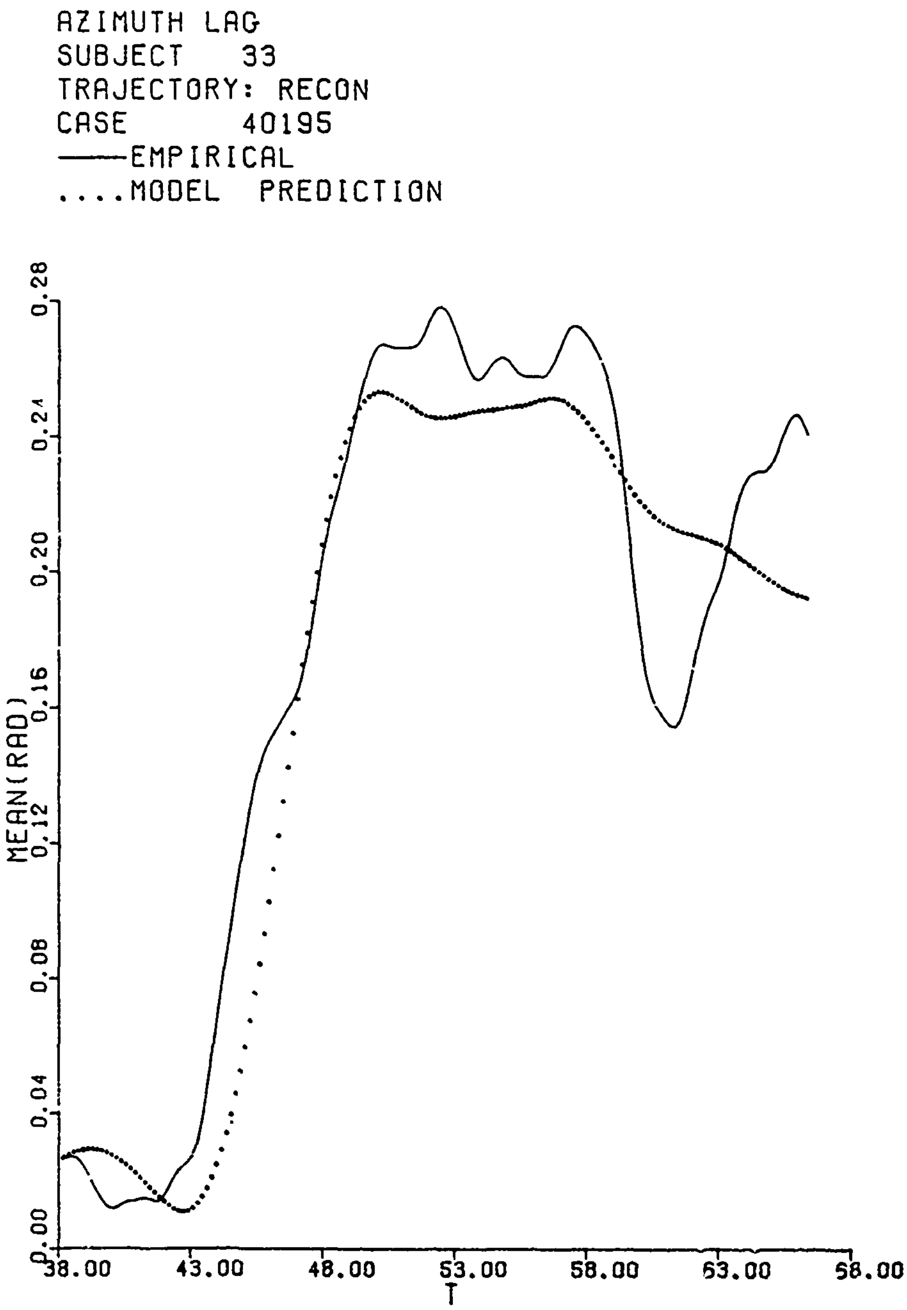


Figure 7a. Mean Tracking Error--Azimuth--Recon

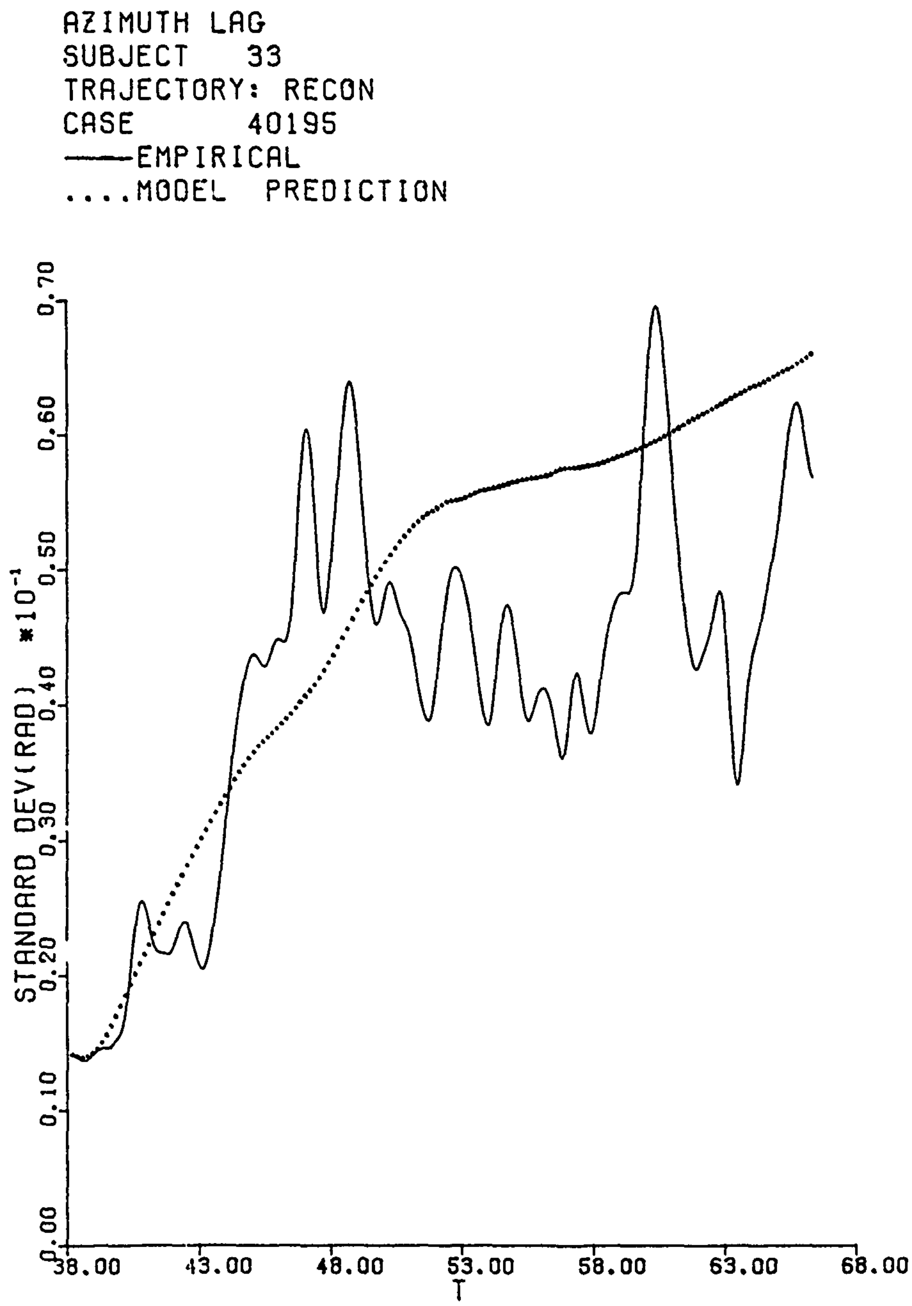


Figure 7b. Standard Deviation of Tracking Error--Azimuth--Recon

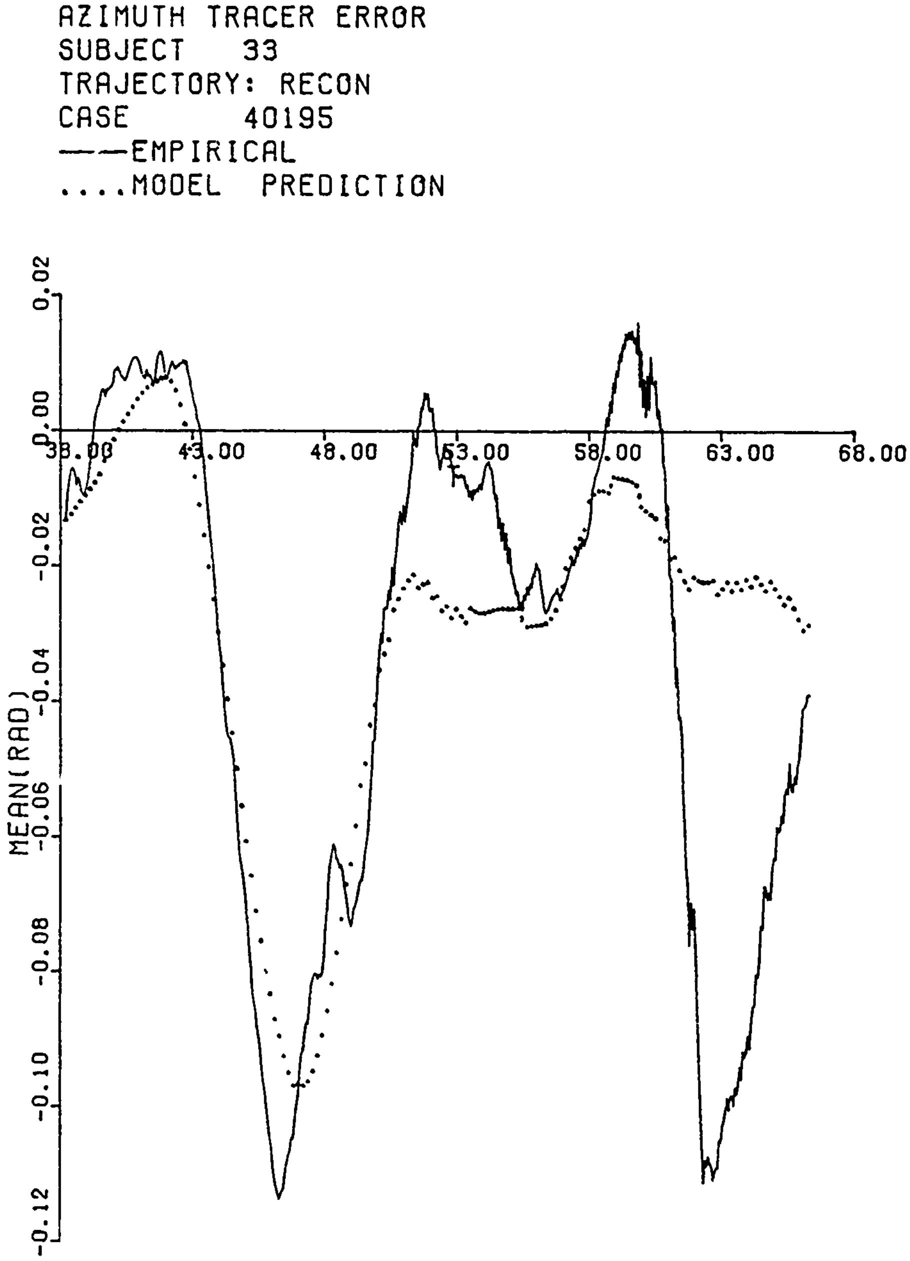


Figure 8a. Mean Tracer Error--Azimuth--Recon

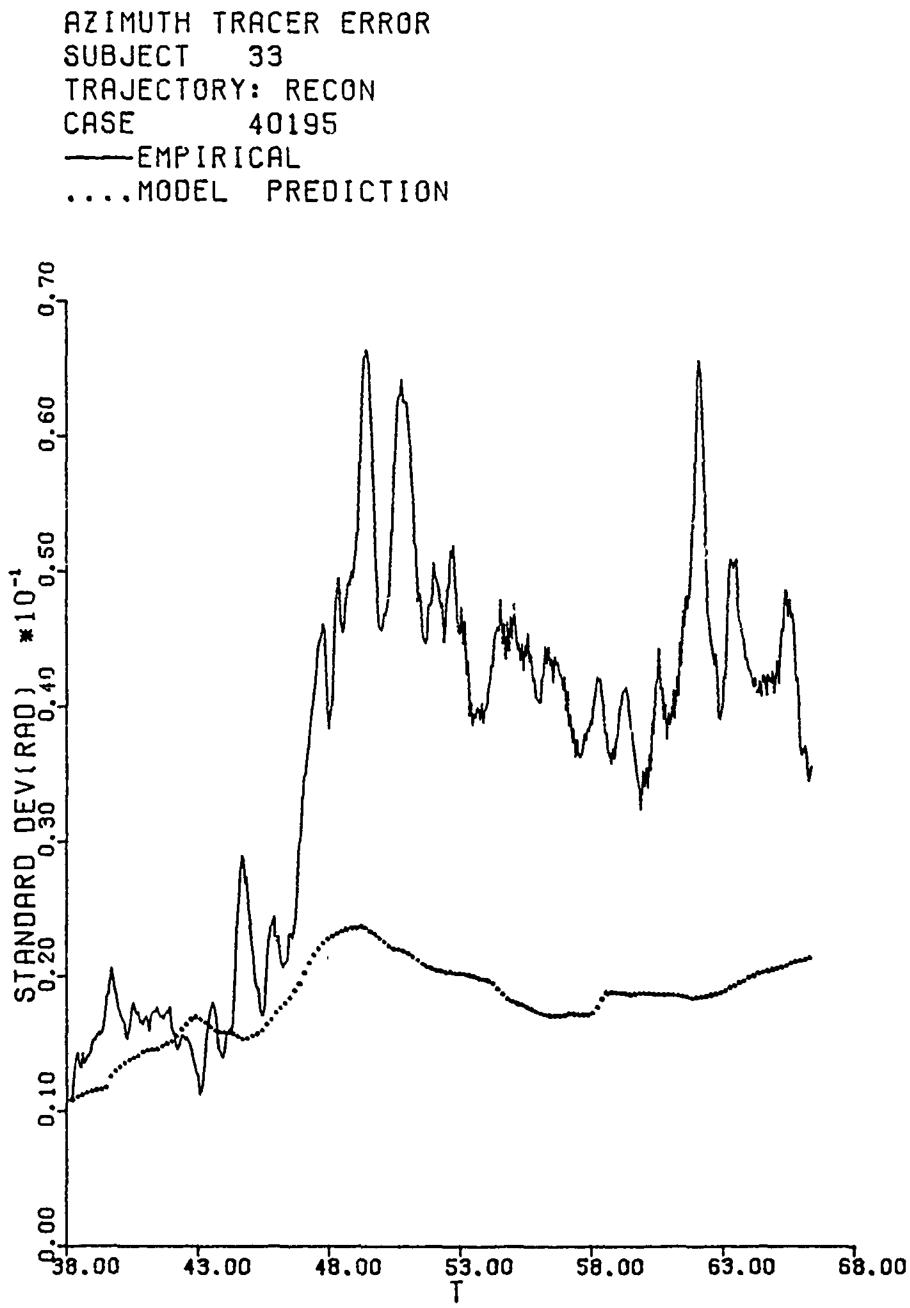


Figure 8b. Standard Deviation of Tracer Error--Azimuth--Recon

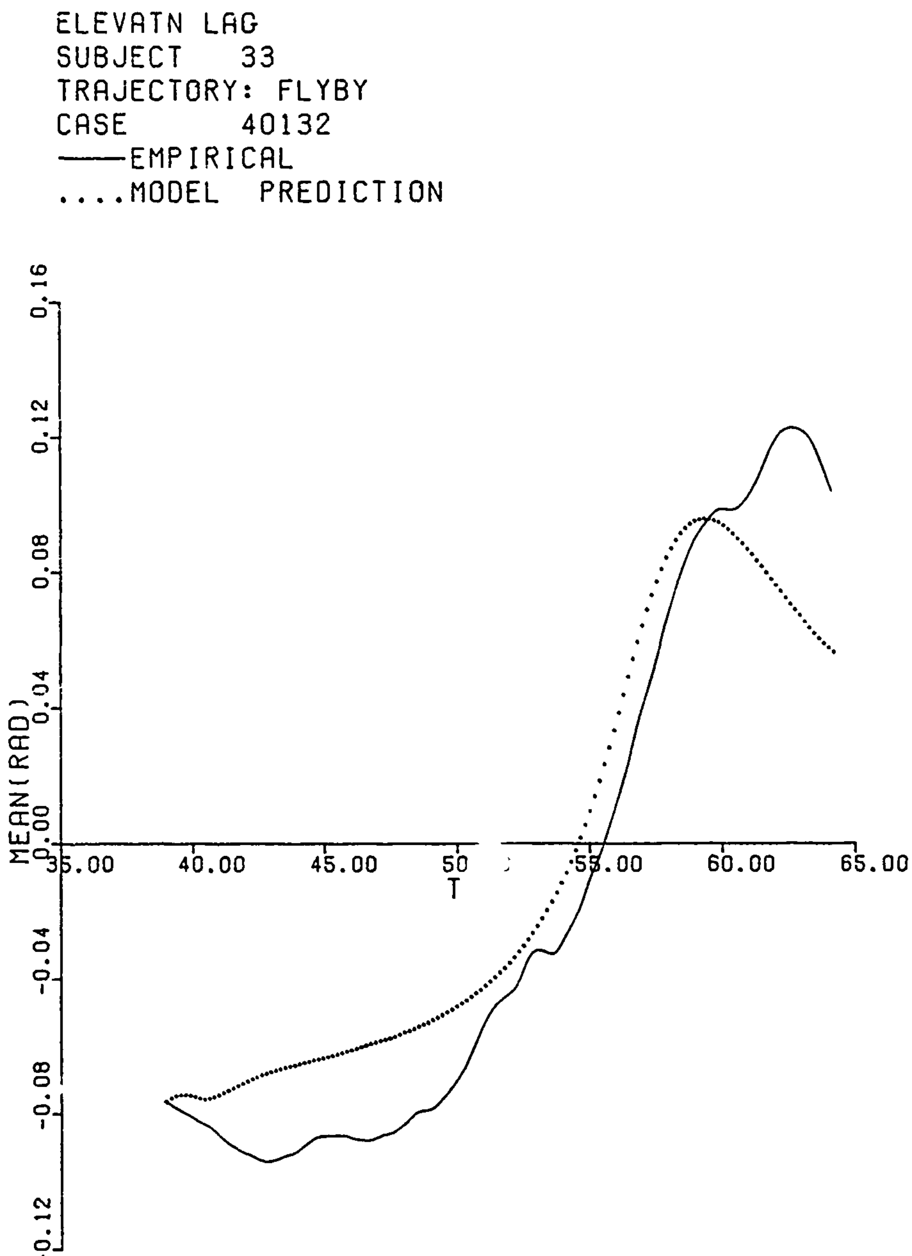


Figure 9a. Mean Tracking Error--Elevation--Flyby

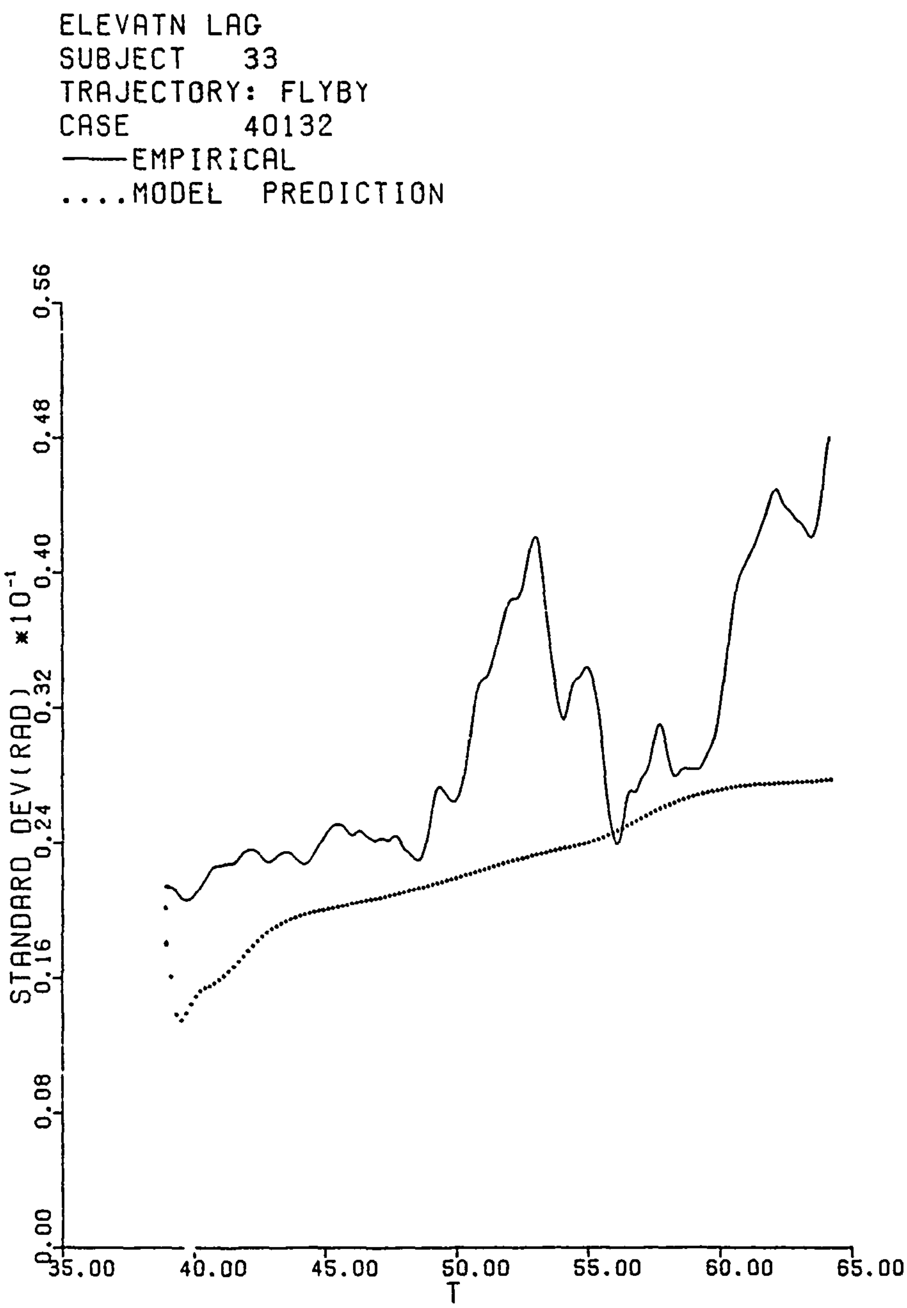


Figure 9b. Standard Deviation of Tracking Error--Elevation--Flyby

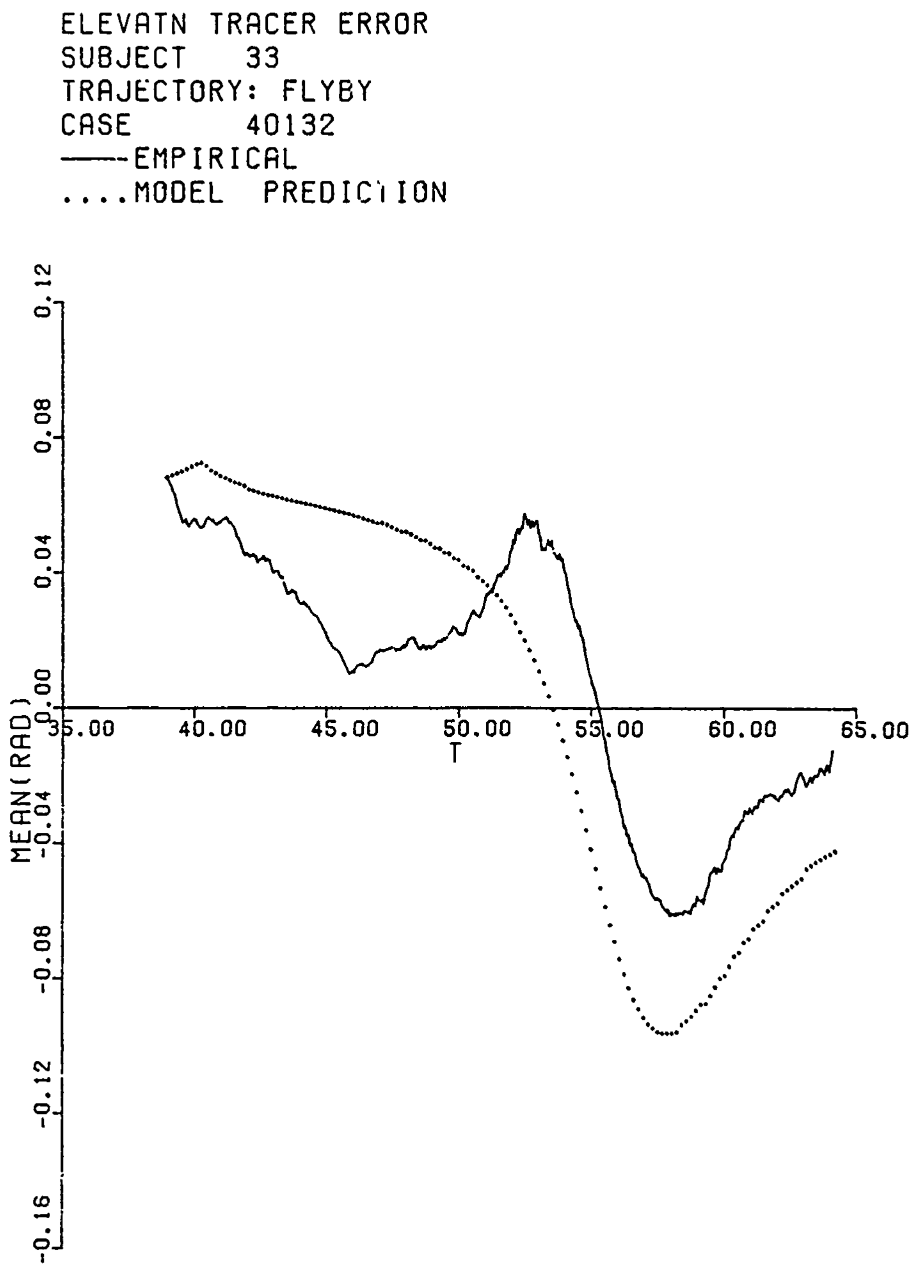


Figure 10a. Mean Tracer Error--Elevation--Flyby

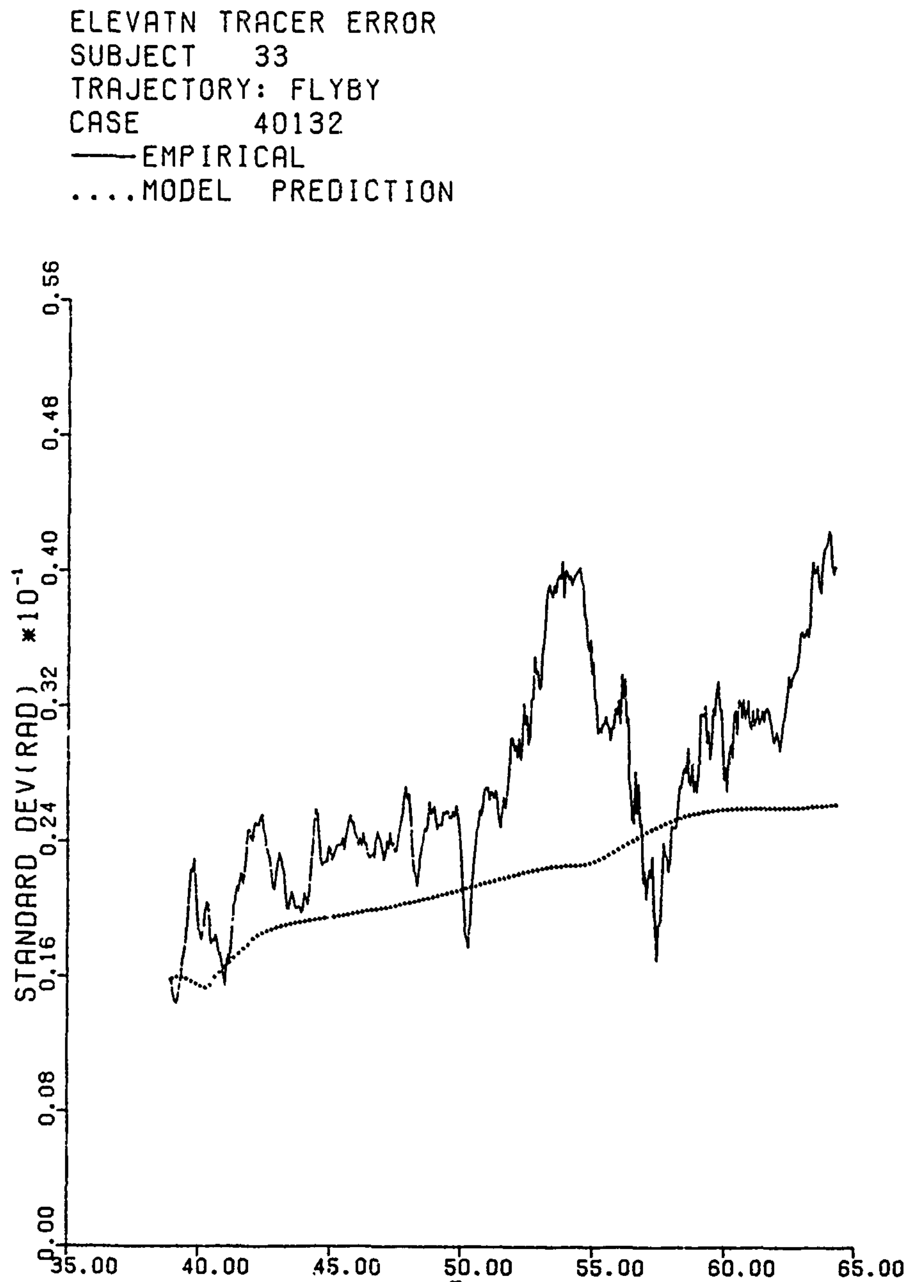


Figure 10b. Standard Deviation of Tracer Error--Elevation--Flyby

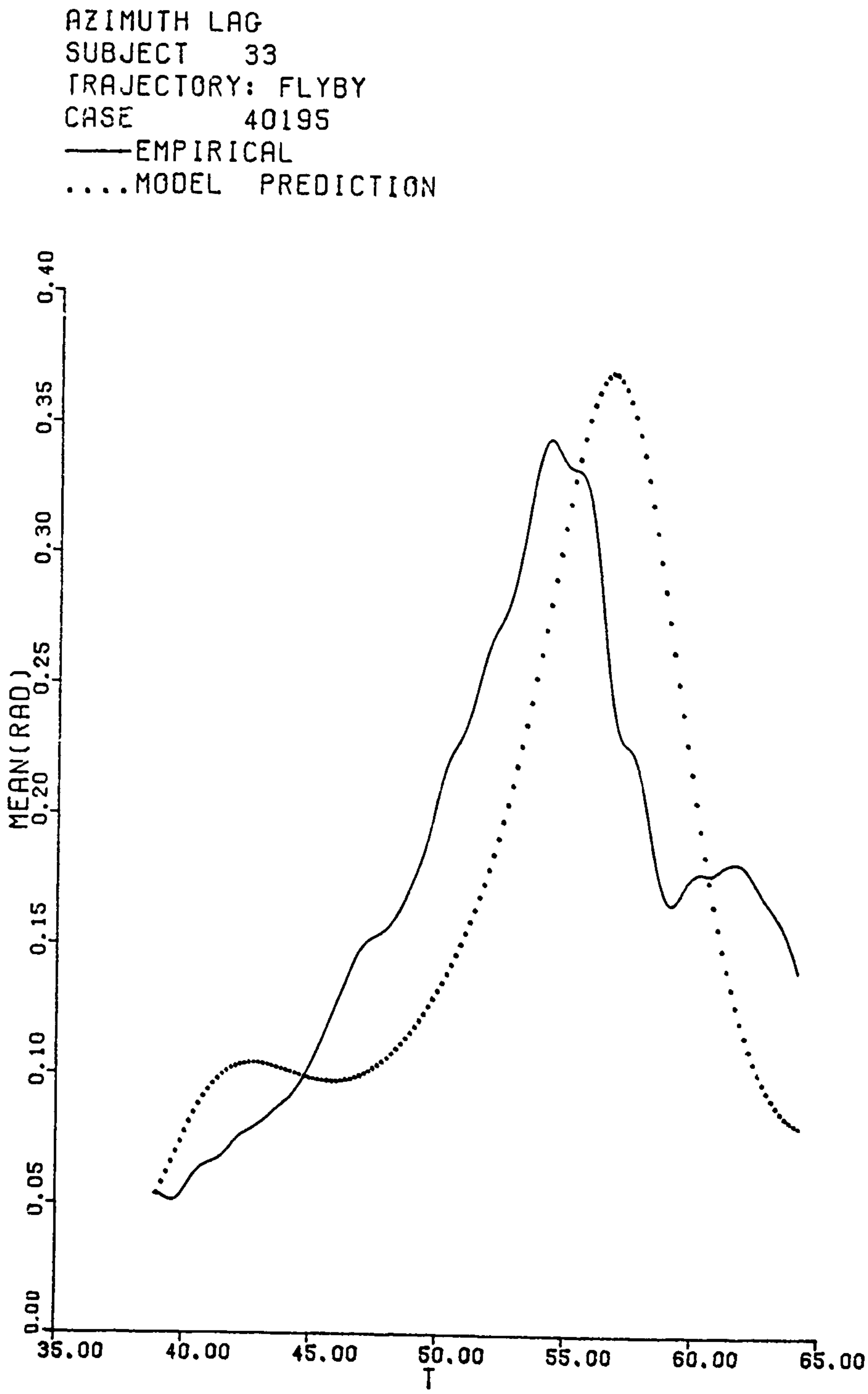


Figure 11a. Mean Tracking Error--Azimuth--Flyby

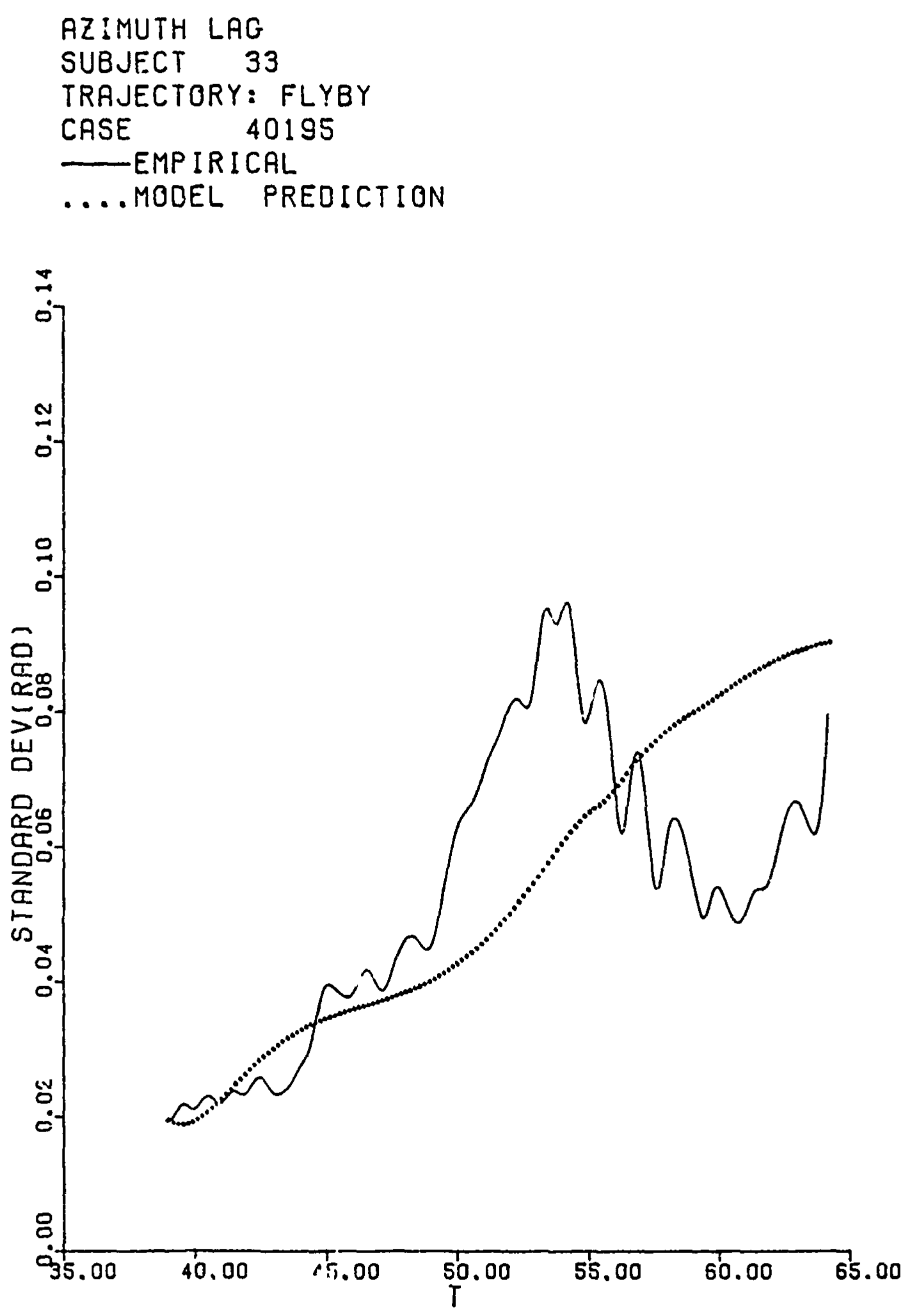


Figure 11b. Standard Deviation of Tracking Error--Azimuth--Flyby

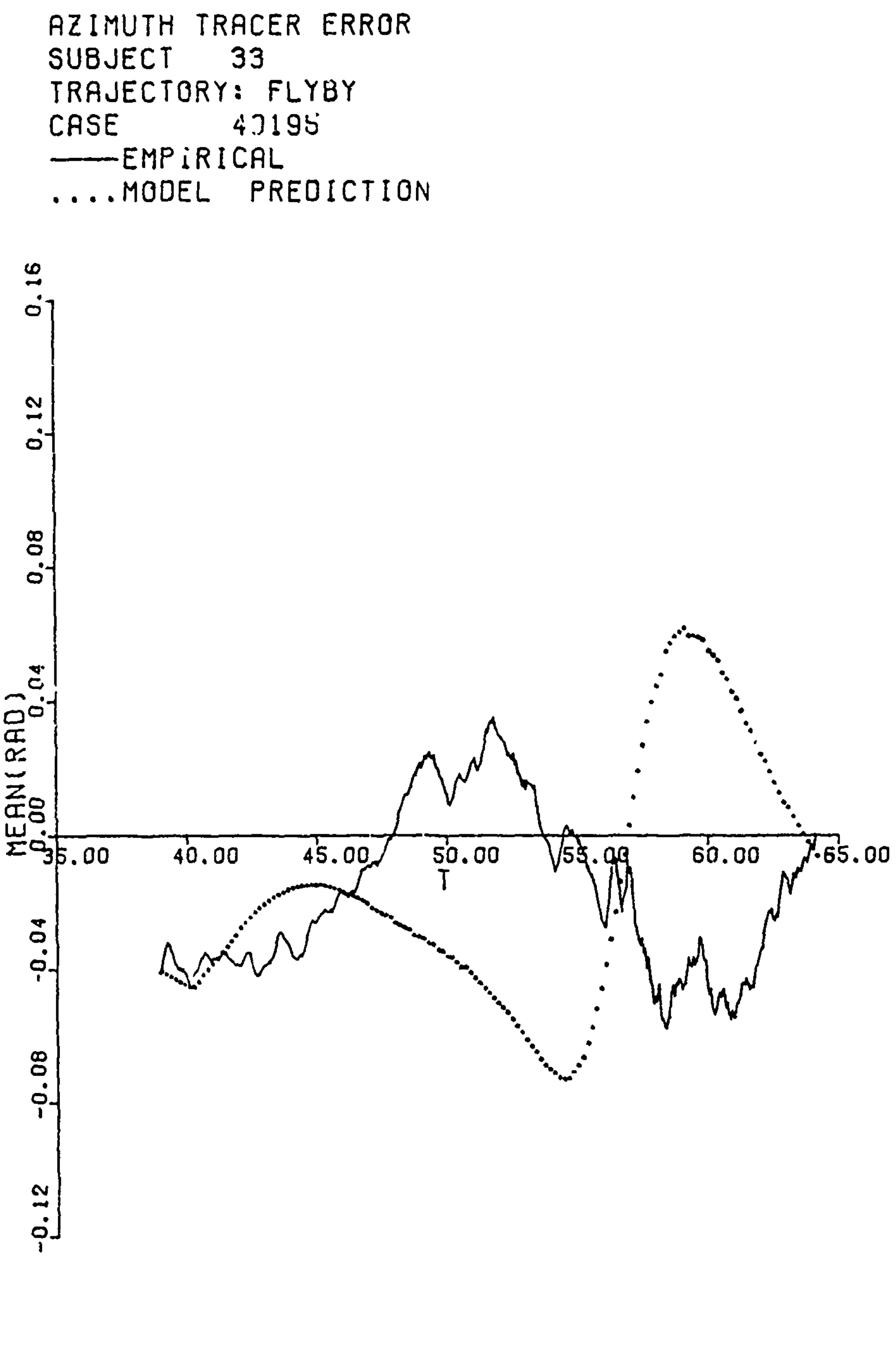


Figure 12a. Mean Tracer Error--Azimuth--Flyby

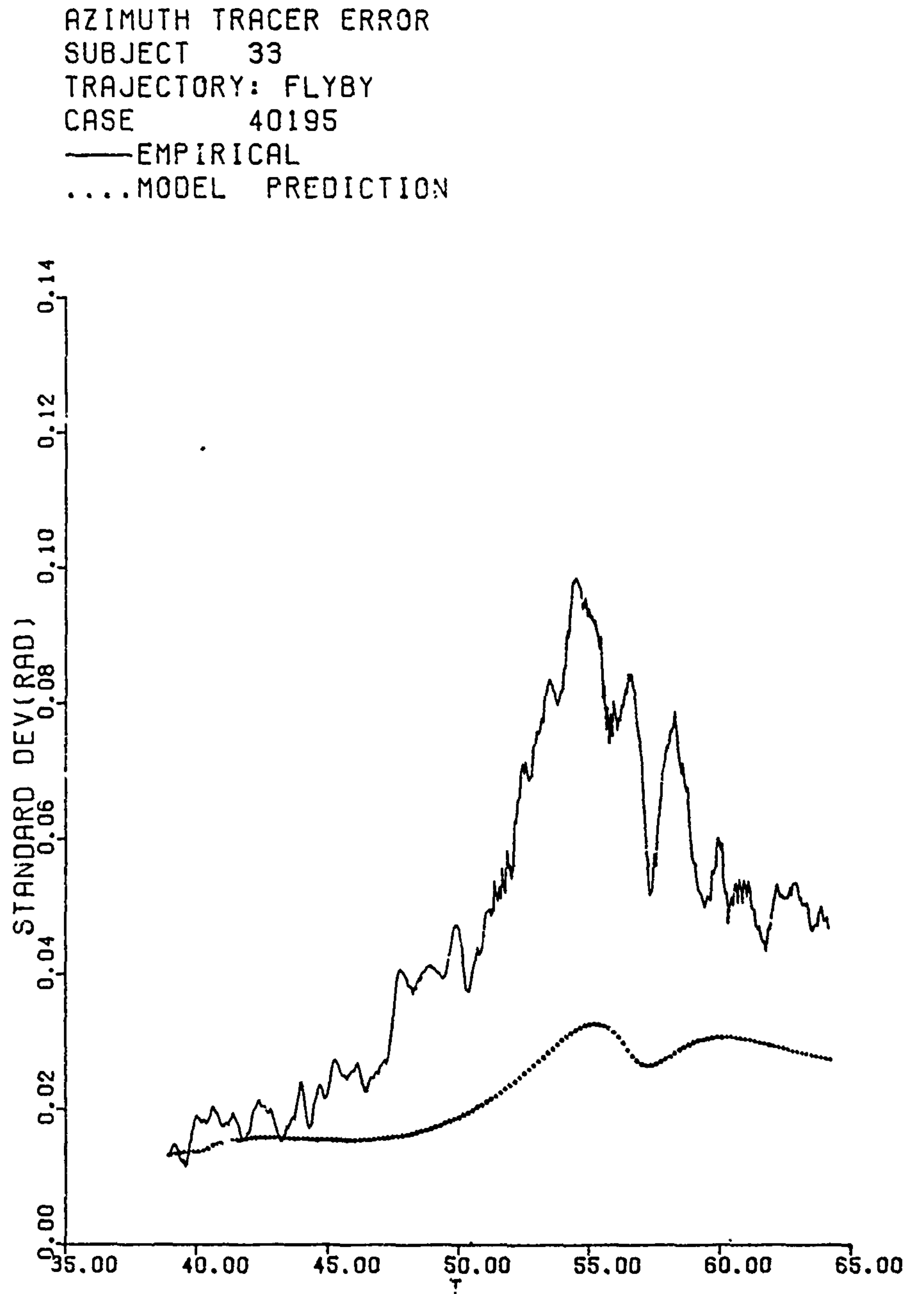


Figure 12b. Standard Deviation of Tracer Error--Azimuth--Flyby

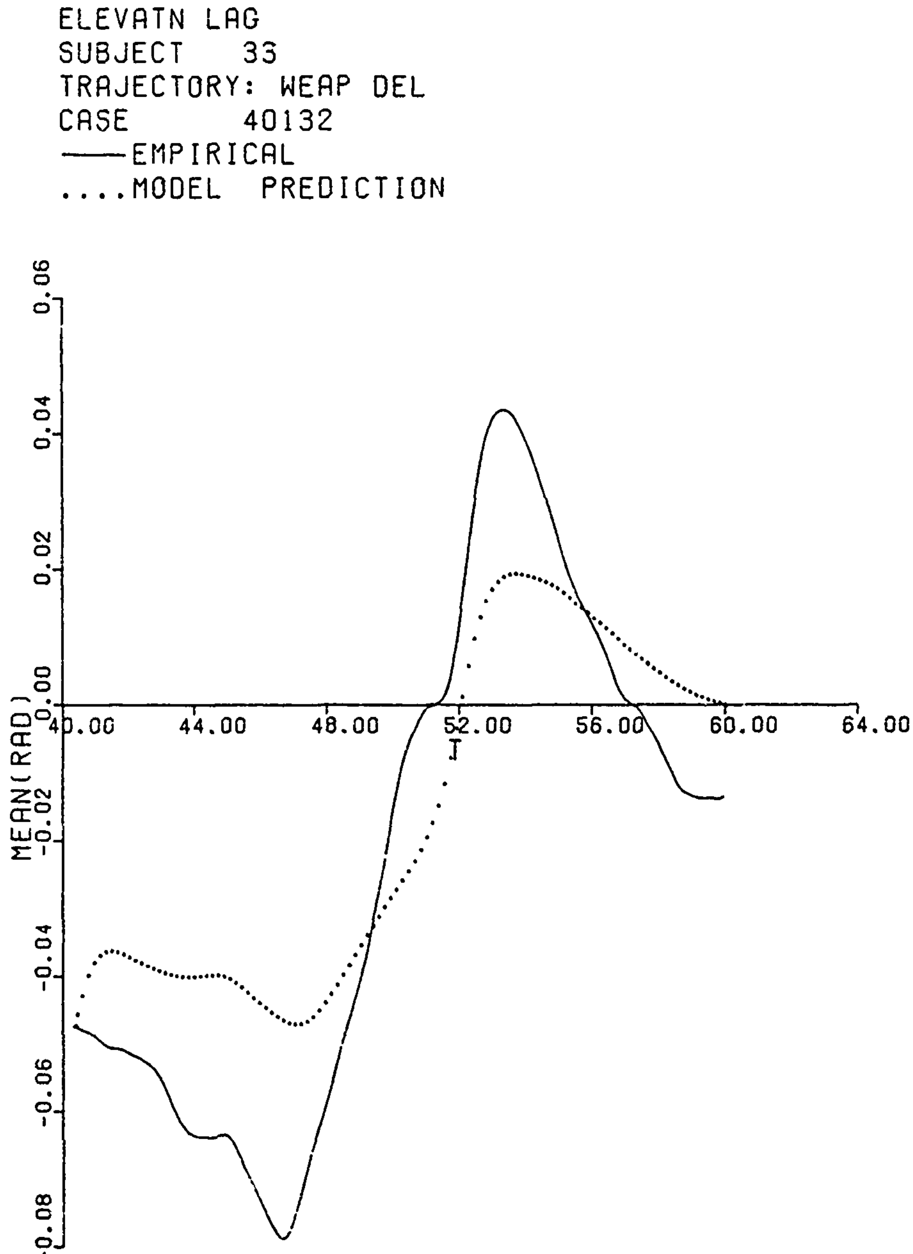


Figure 13a. Mean Tracking Error--Elevation--Weapon Delivery

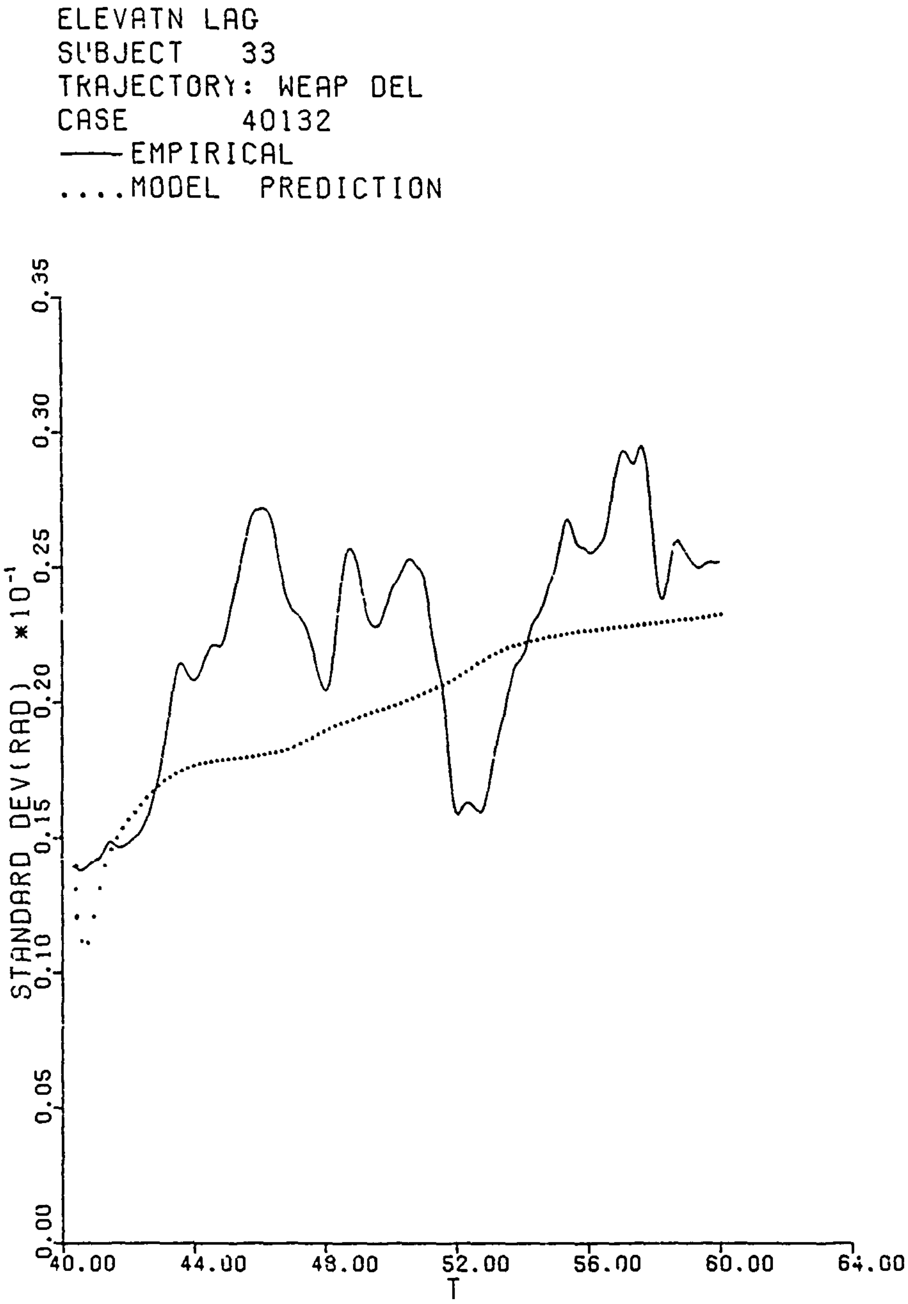


Figure 13b. Standard Deviation of Tracking Error--Elevation--Weapon Delivery

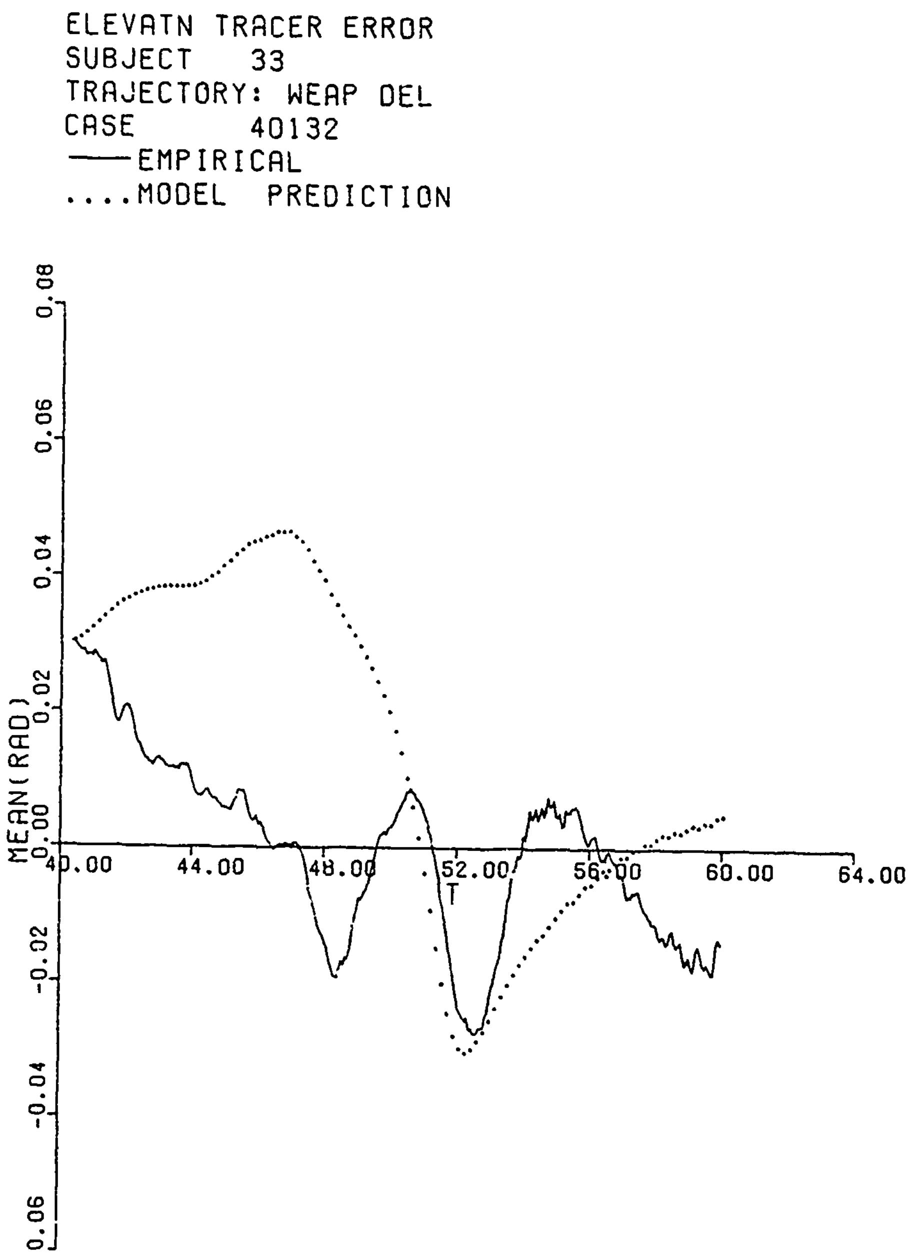


Figure 14a. Mean Tracer Error--Elevation--Weapon Delivery

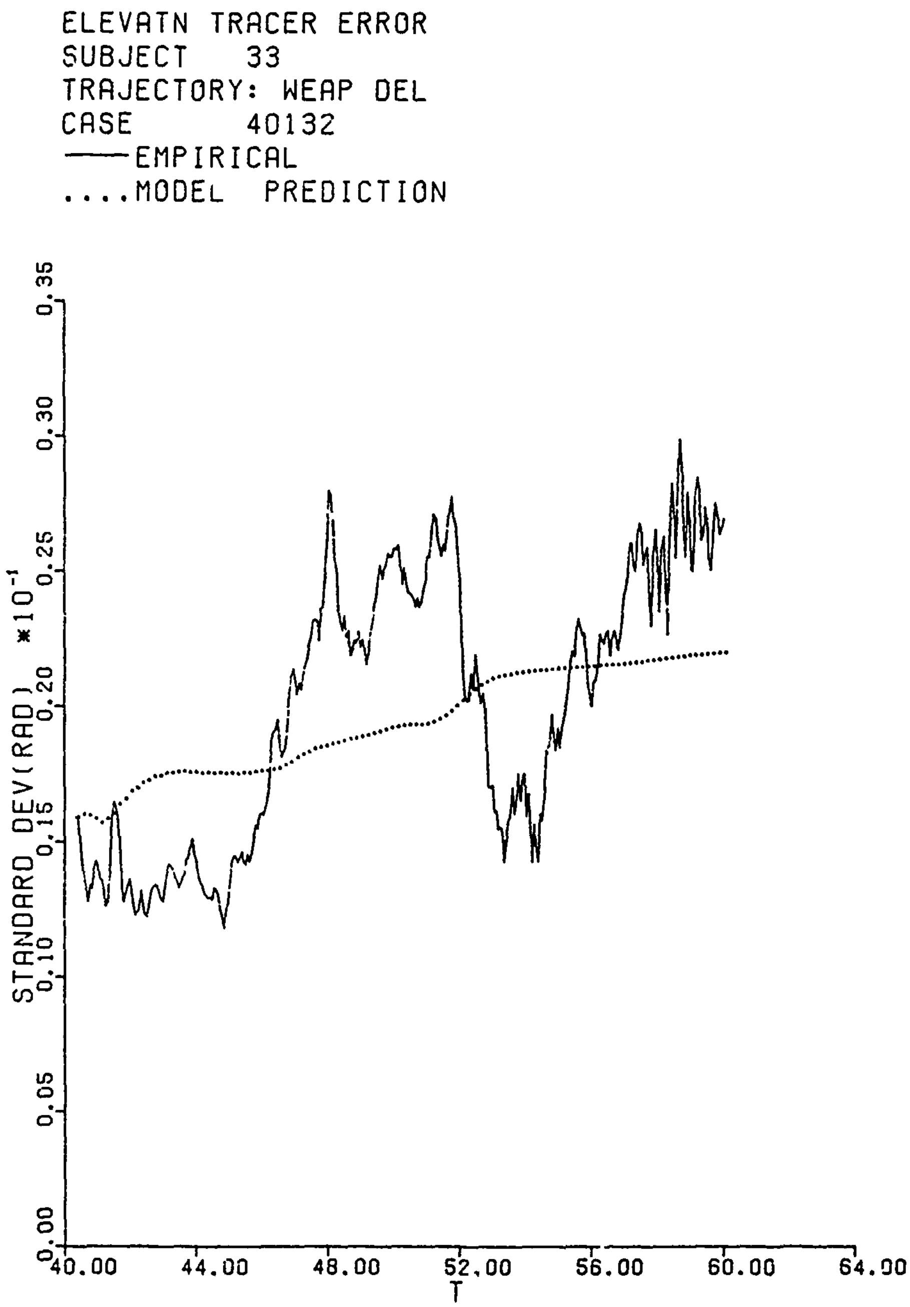


Figure 14b. Standard Deviation of Tracer Error—Elevation—Weapon Delivery

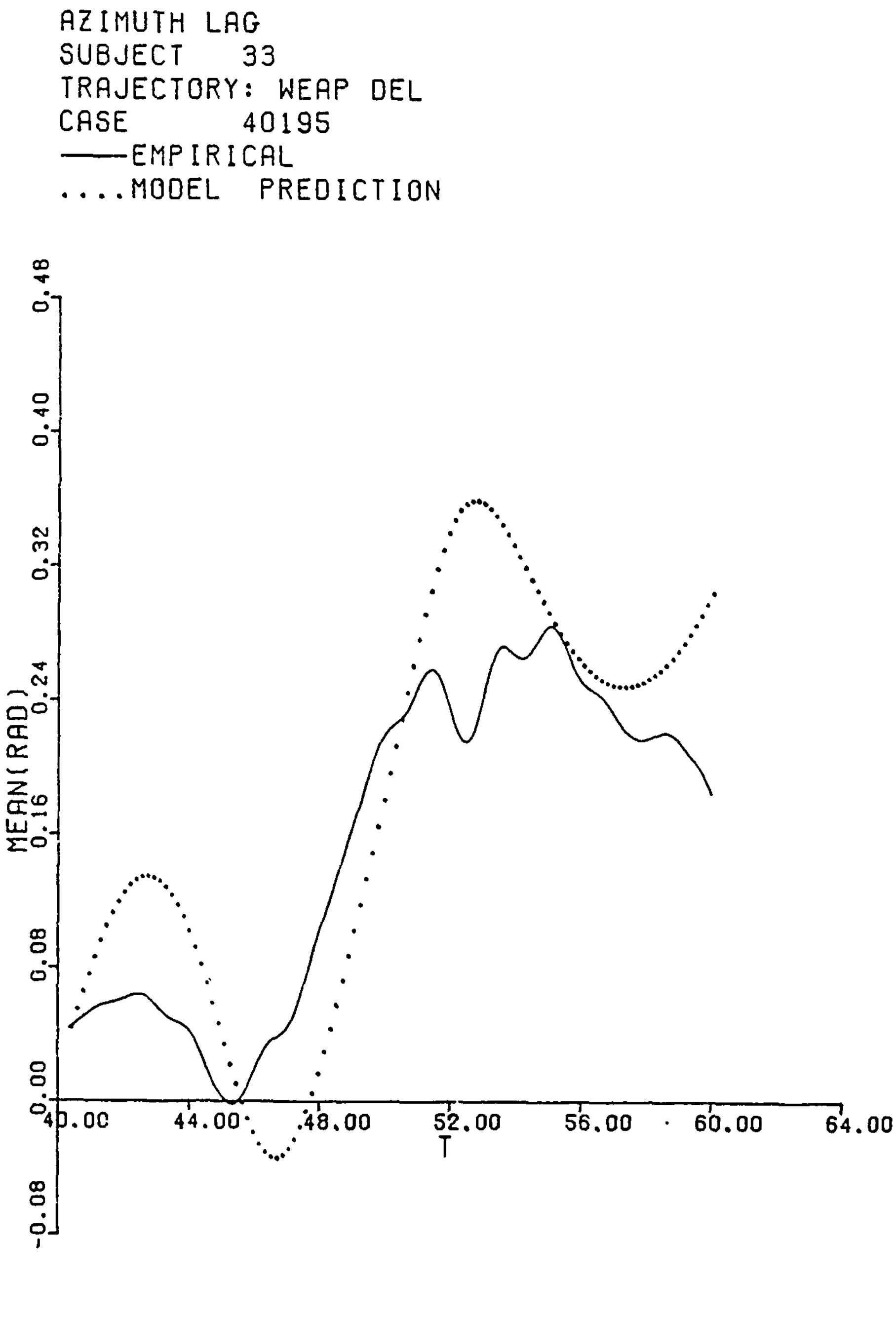


Figure 15a. Mean Tracking Error--Azimuth--Weapon Delivery

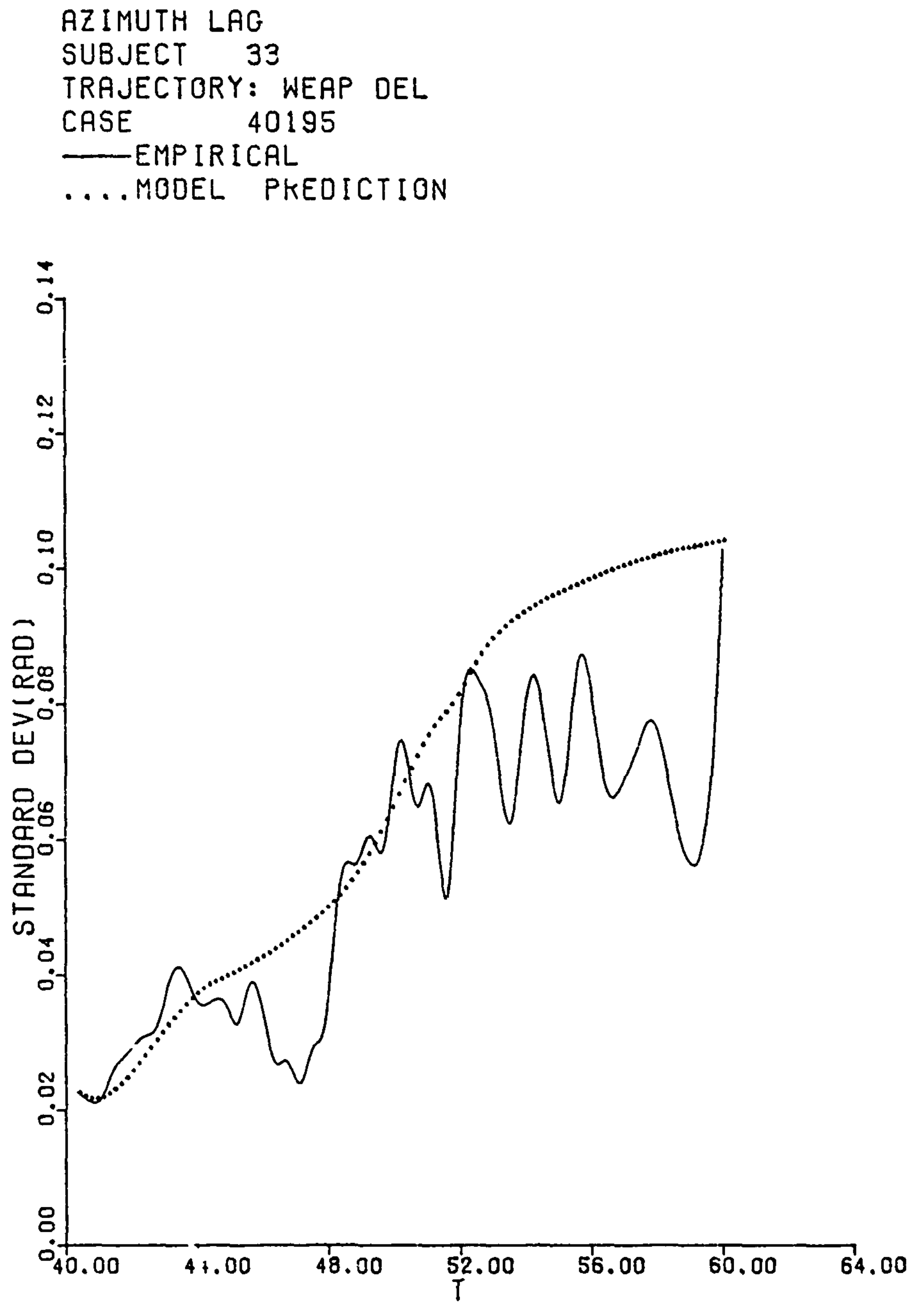


Figure 15b. Standard Deviation of Tracking Error—Azimuth—Weapon Delivery

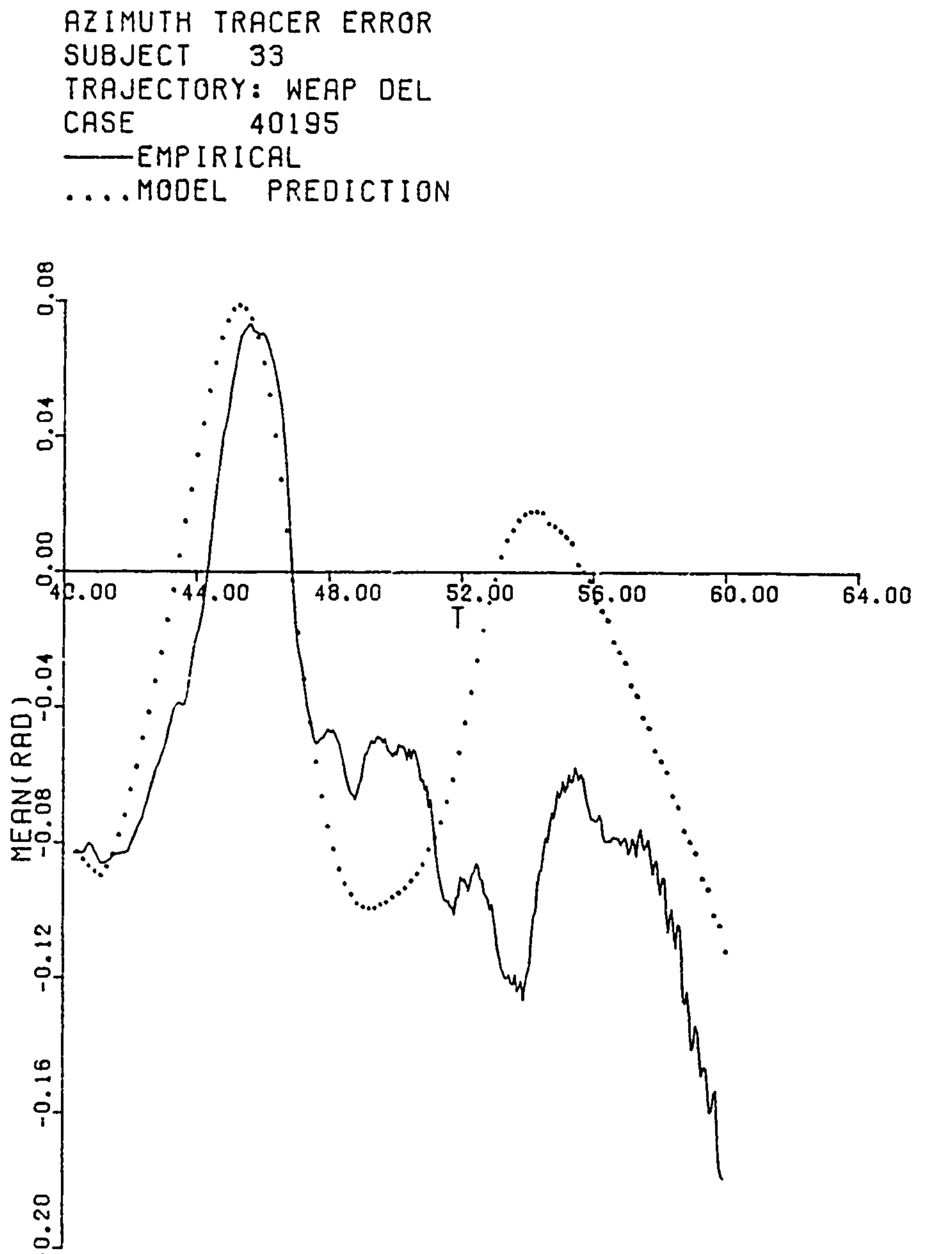


Figure 16a. Mean Tracer Error--Azimuth--Weapon Delivery

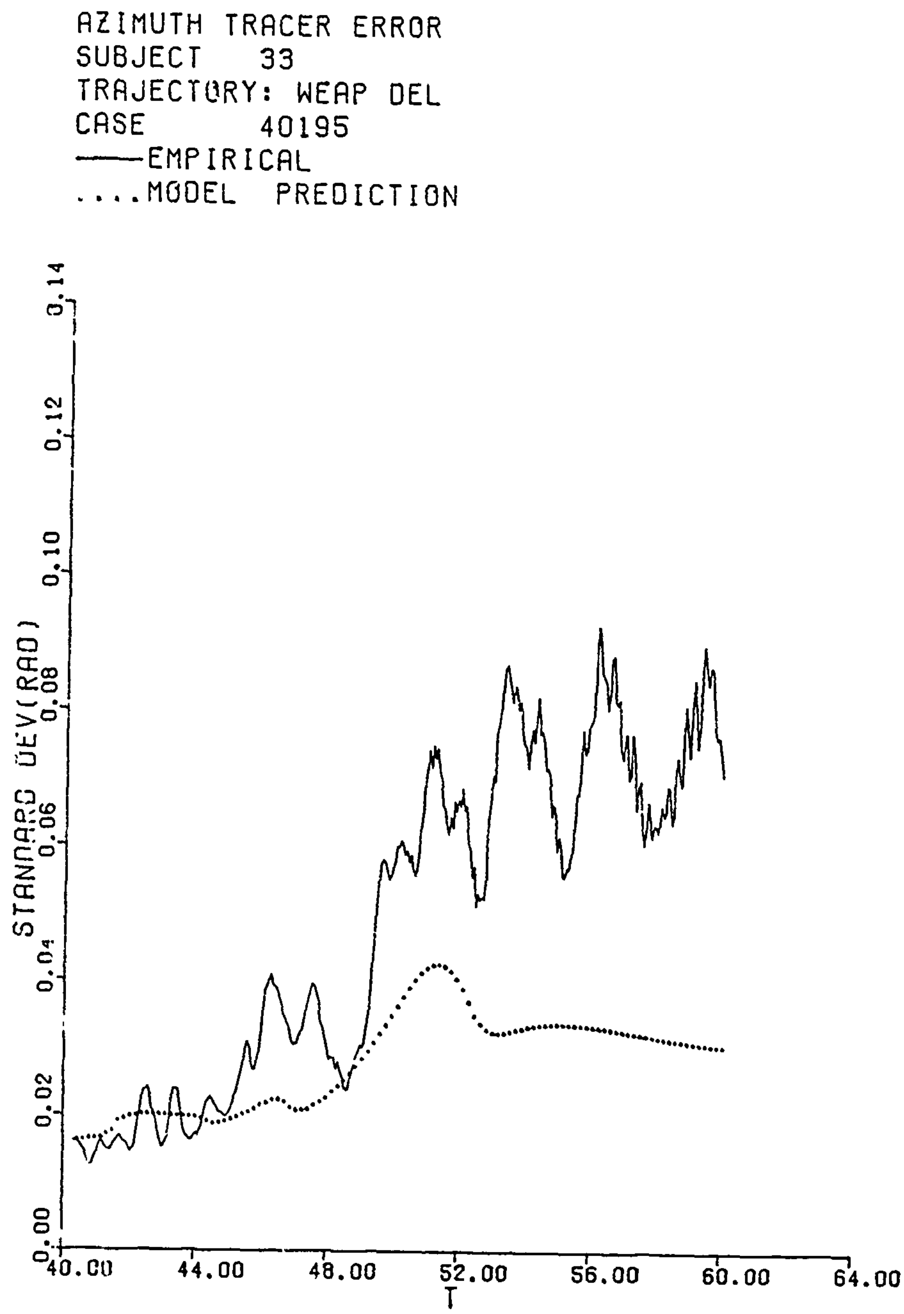


Figure 16b. Standard Deviation of Tracer Error--Azimuth--Weapon Delivery

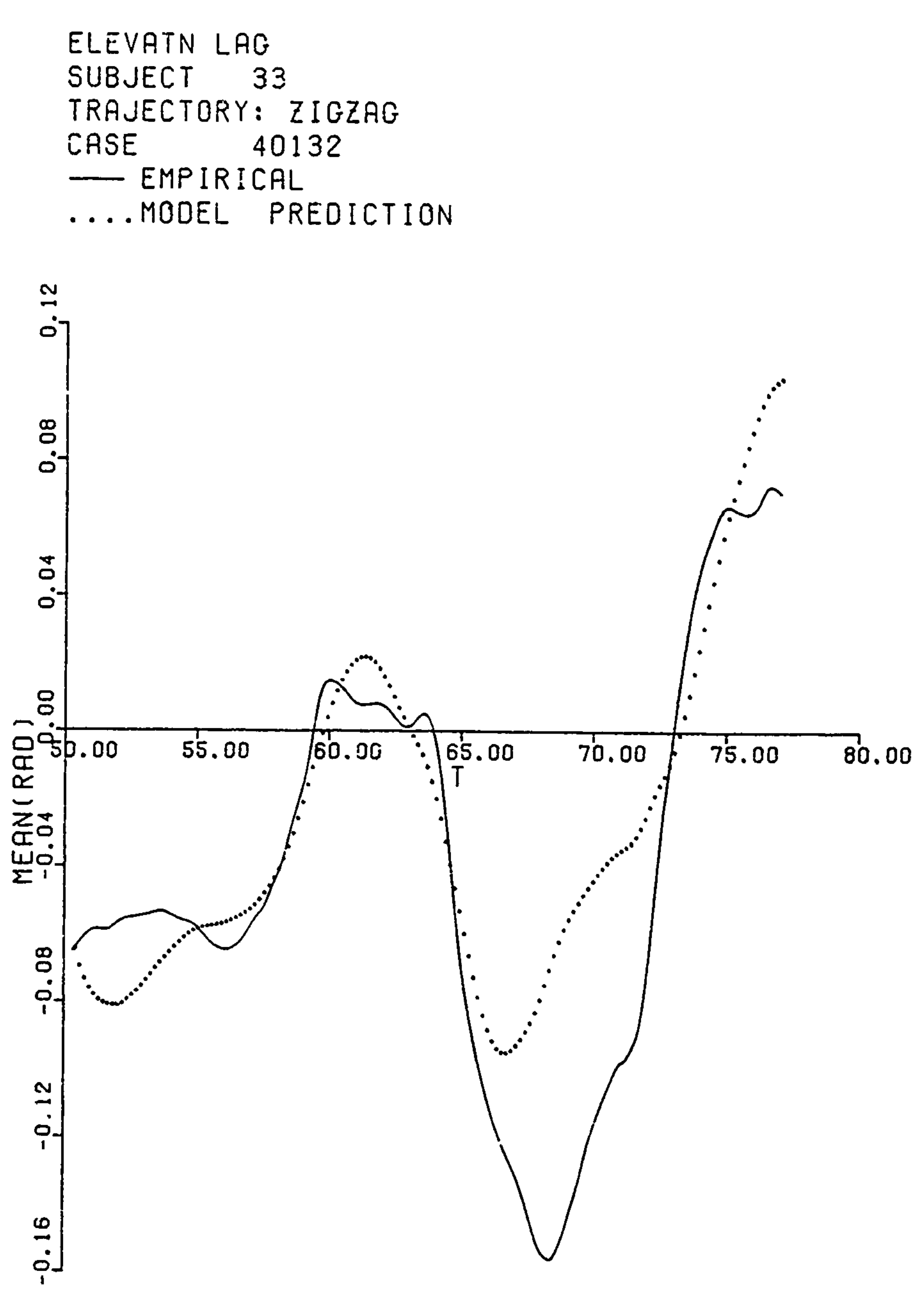


Figure 17a. Mean Tracking Error--Elevation--Zigzag

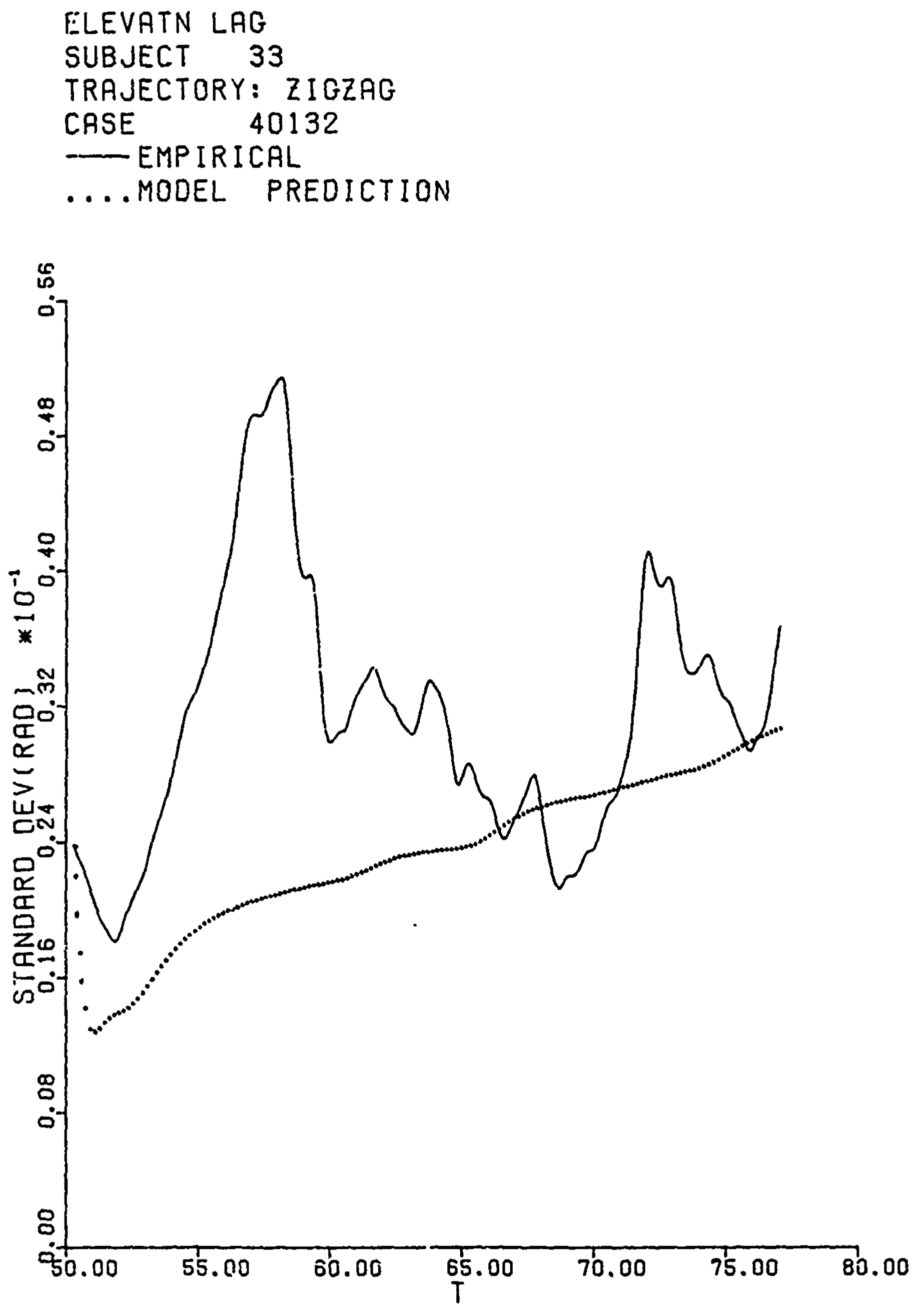


Figure 17b. Standard Deviation of Tracking Error--Elevation--Zigzag

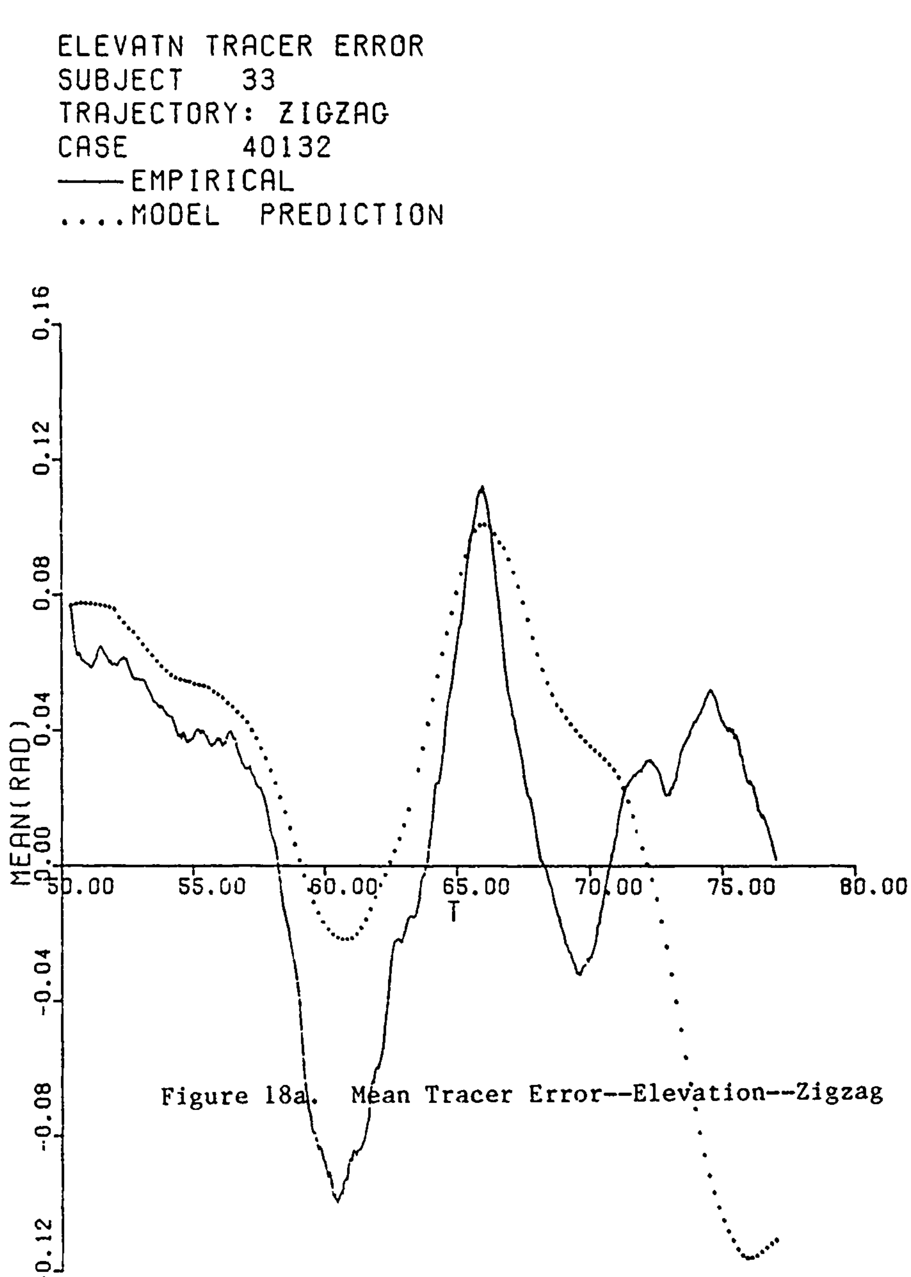


Figure 18a. Mean Tracer Error--Elevation--Zigzag

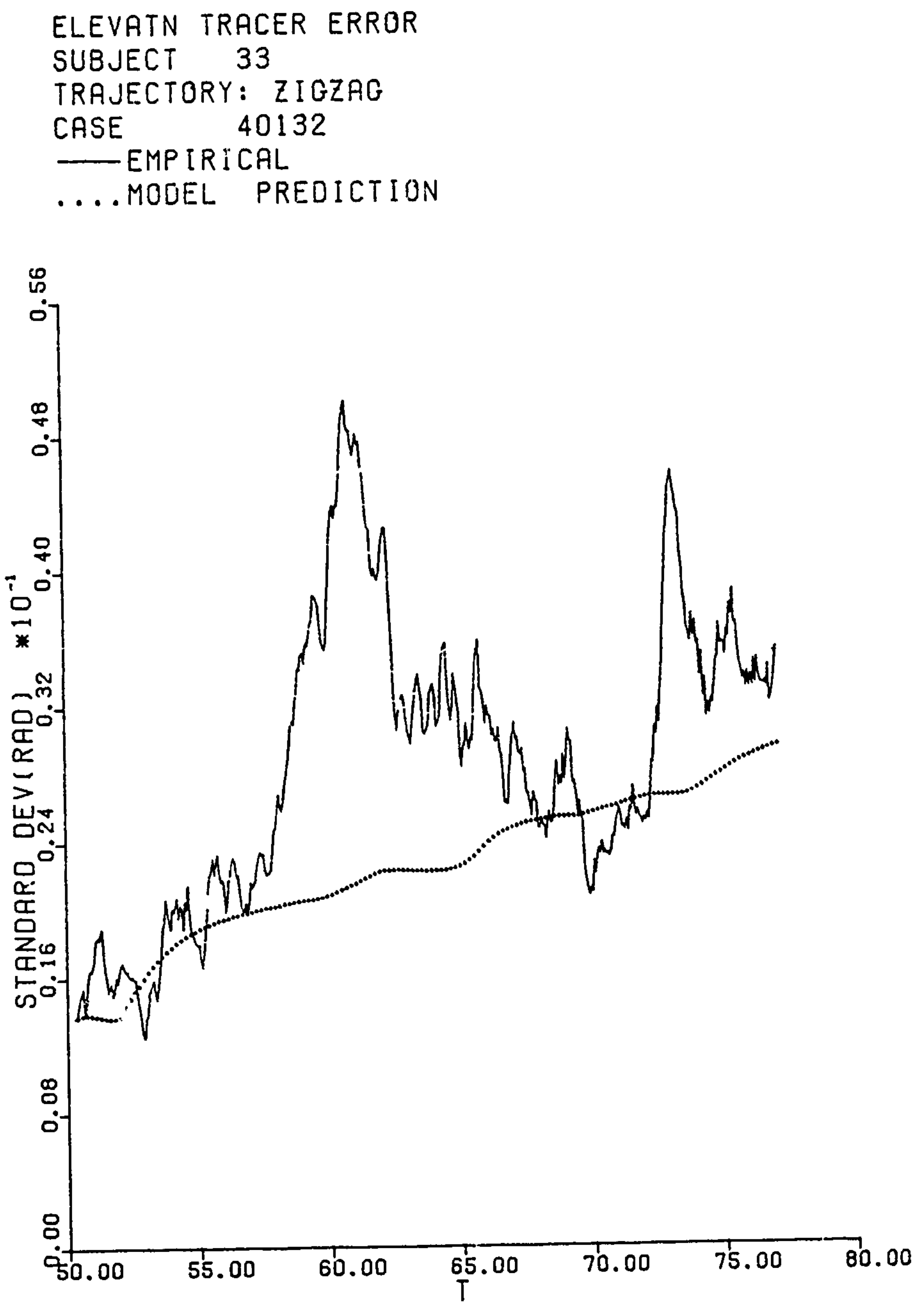


Figure 18b. Standard Deviation of Tracer Error--Elevation--Zigzag

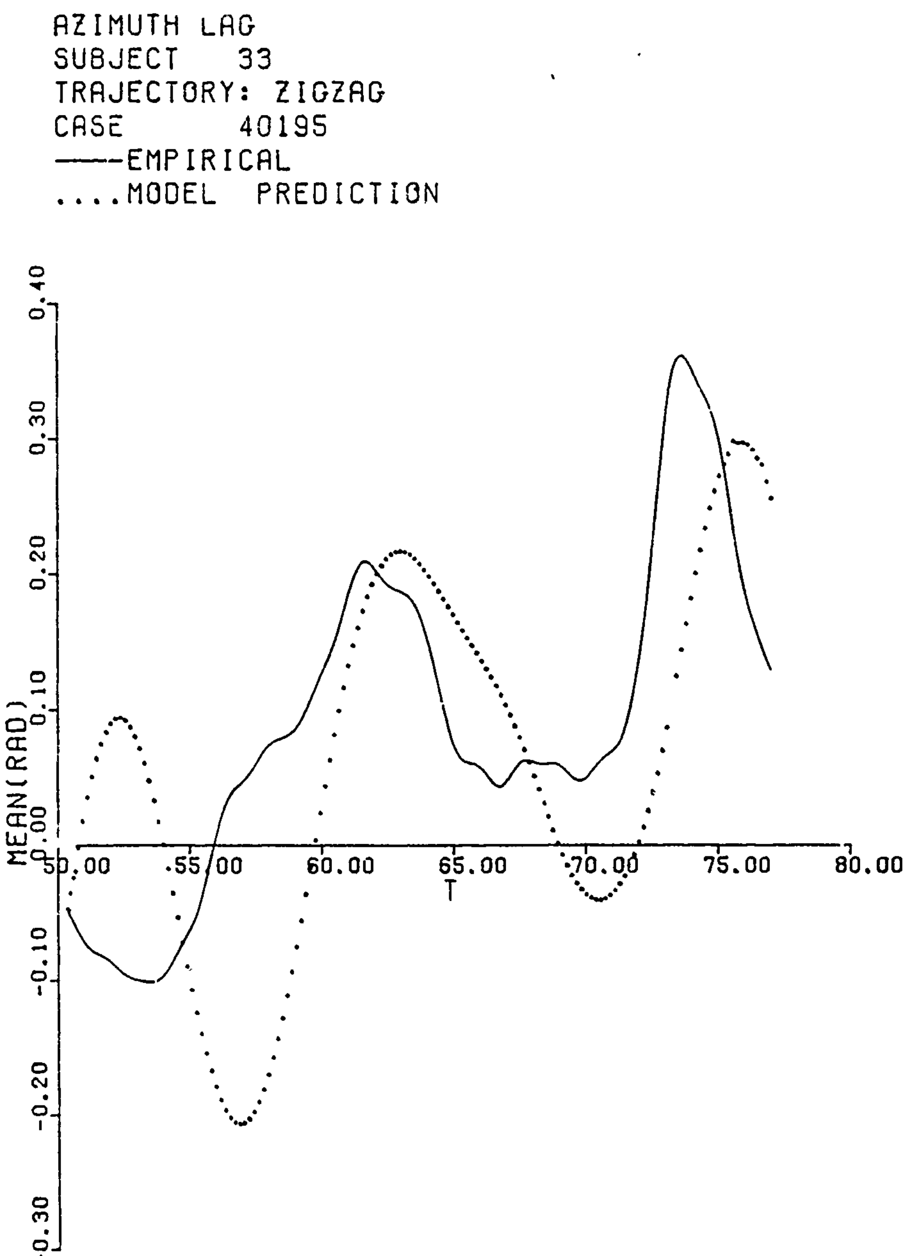


Figure 19a. Mean Tracking Error--Azimuth--Zigzag

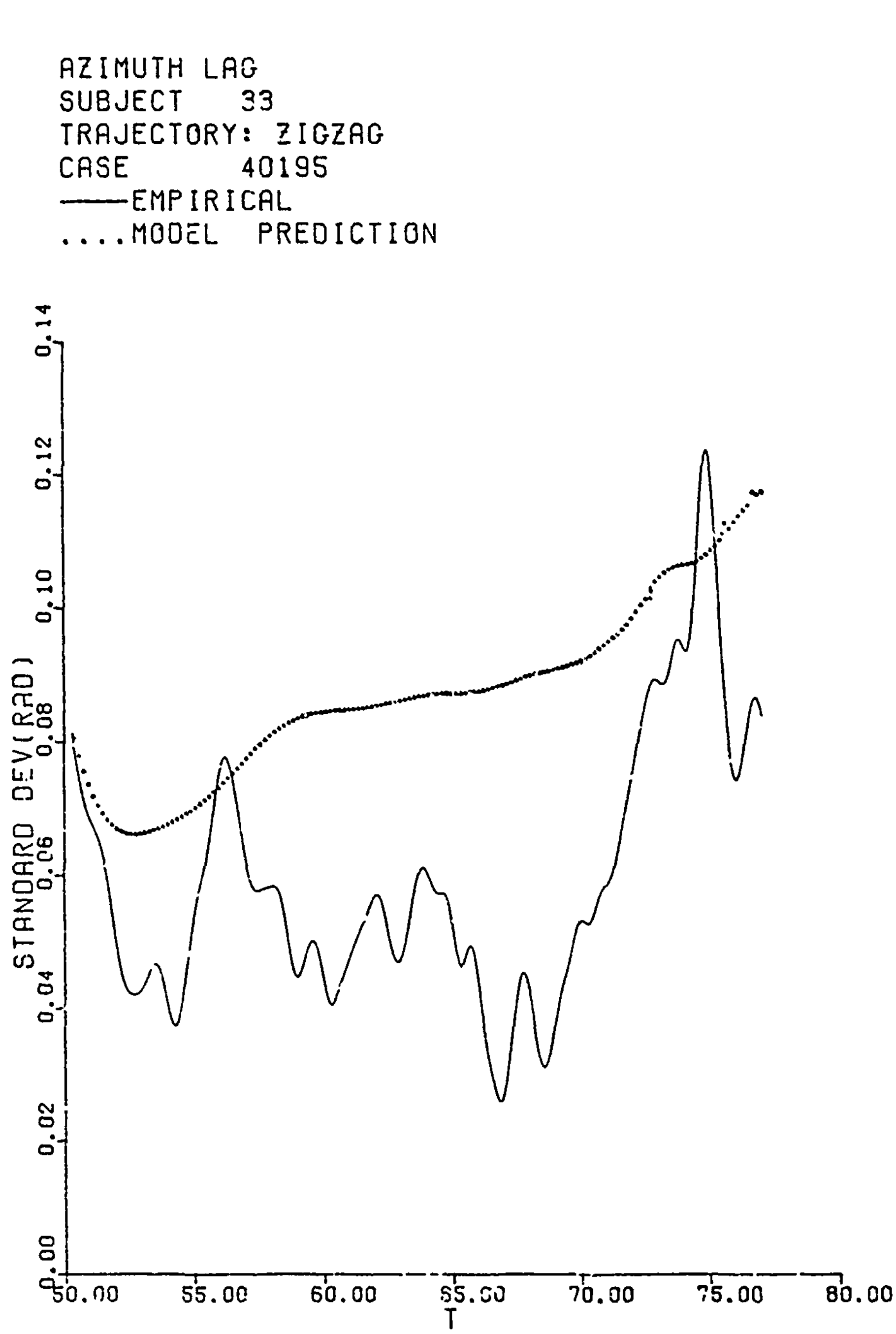


Figure 19b. Standard Deviation of Tracking Error--Azimuth--Zigzag

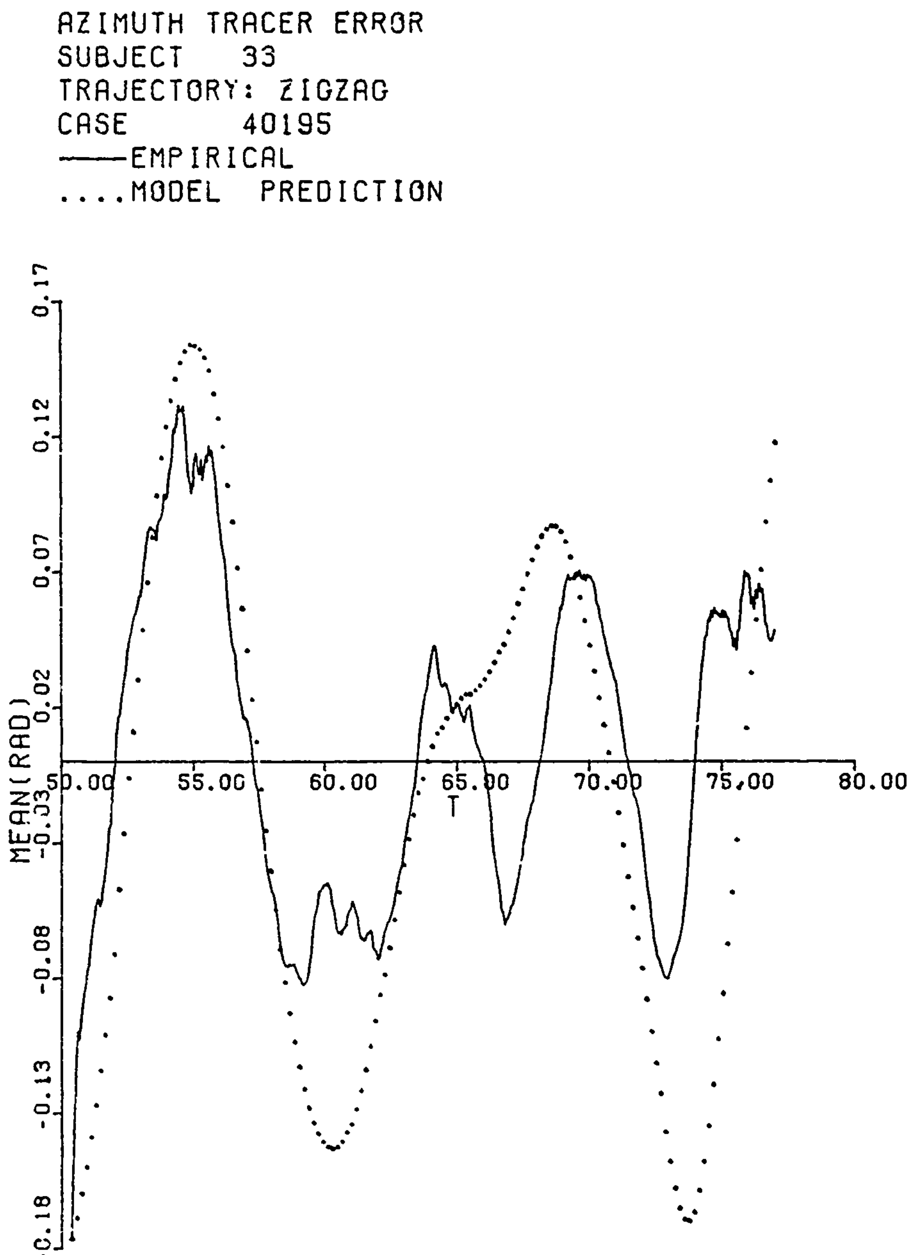


Figure 20a. Mean Tracer Error--Azimuth--Zigzag

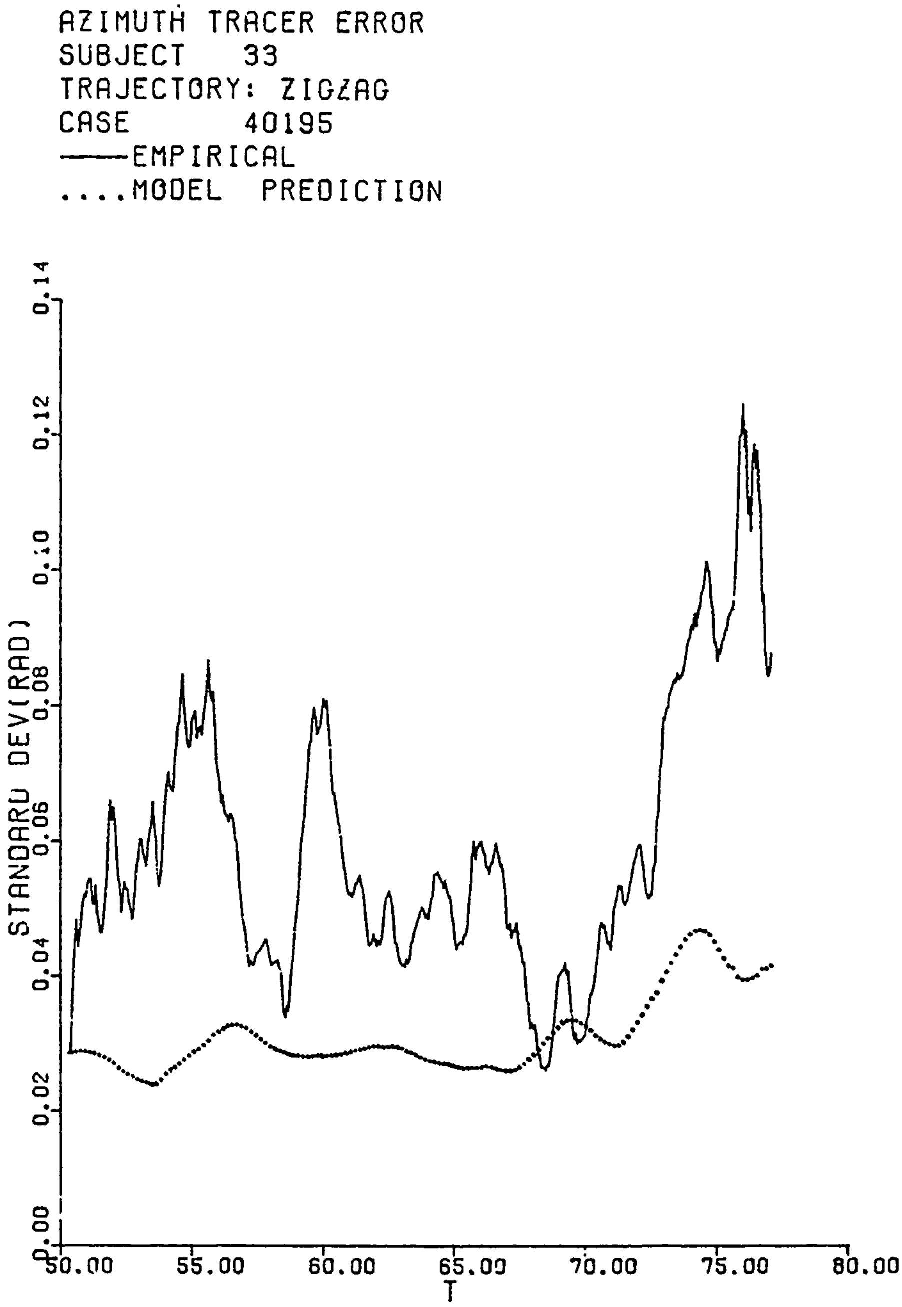


Figure 20b. Standard Deviation of Tracer Error--Azimuth--Zigzag

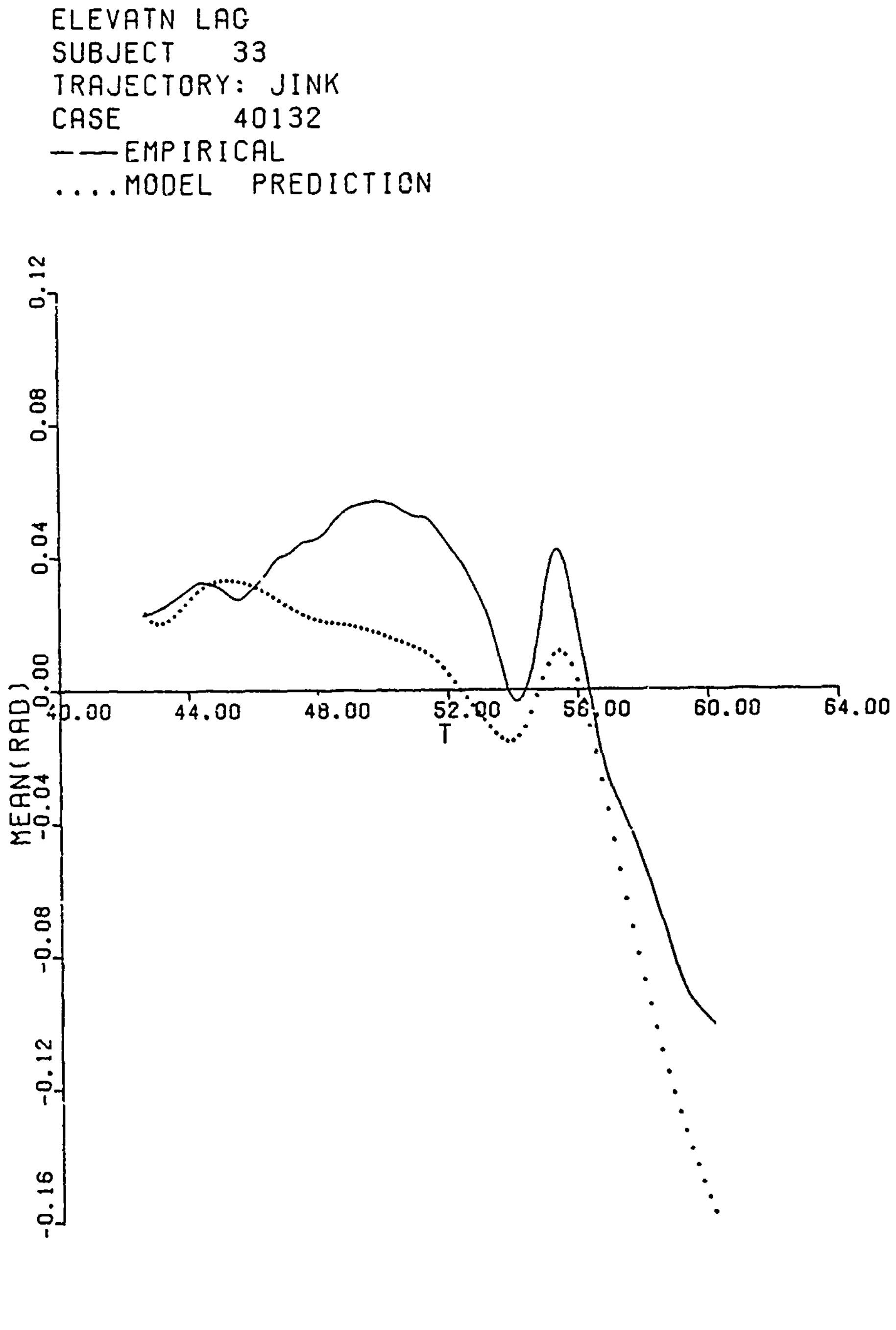


Figure 21a. Mean Tracking Error--Elevation--Jink

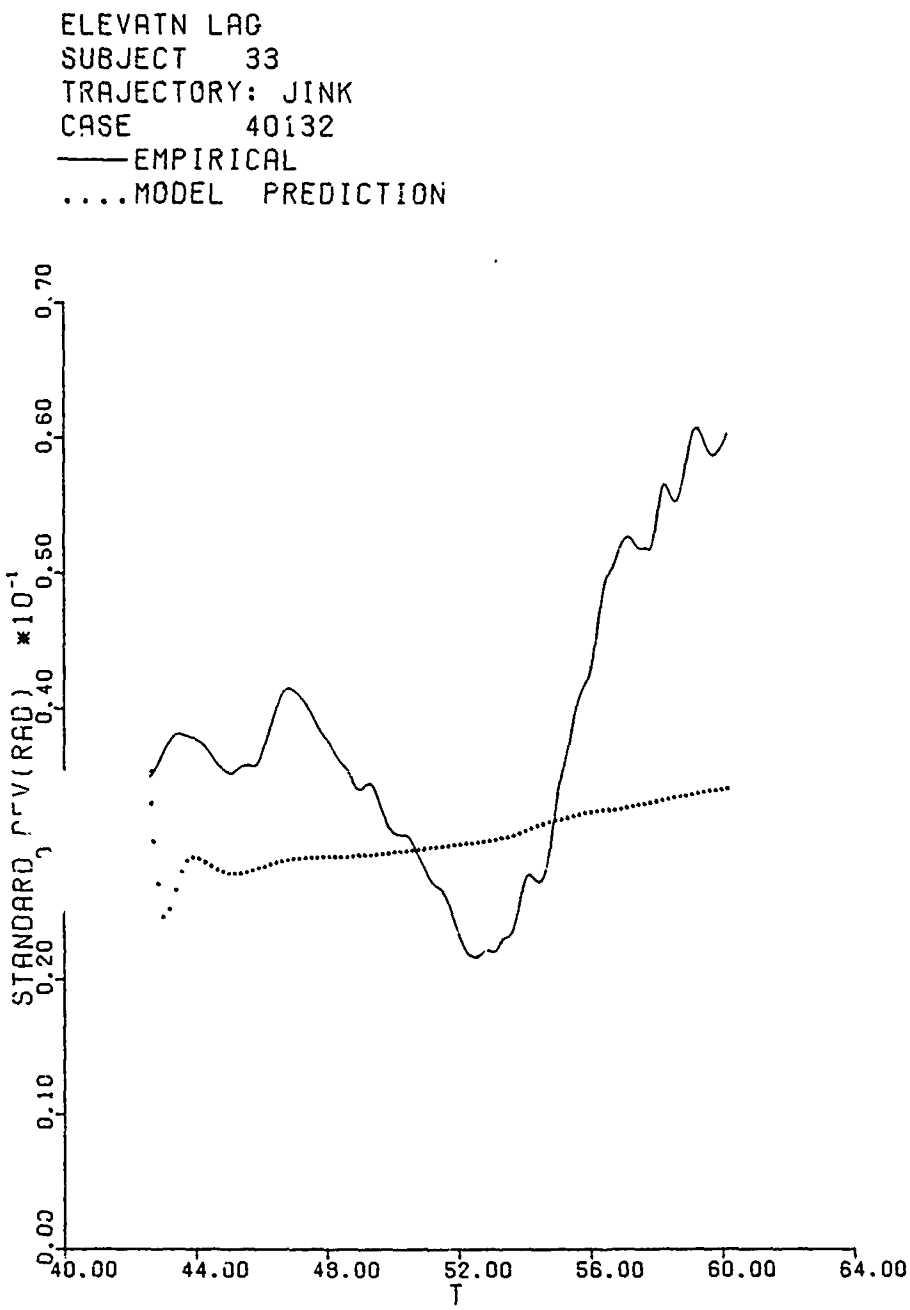


Figure 21b. Standard Deviation of Tracking Error--Elevation--Jink

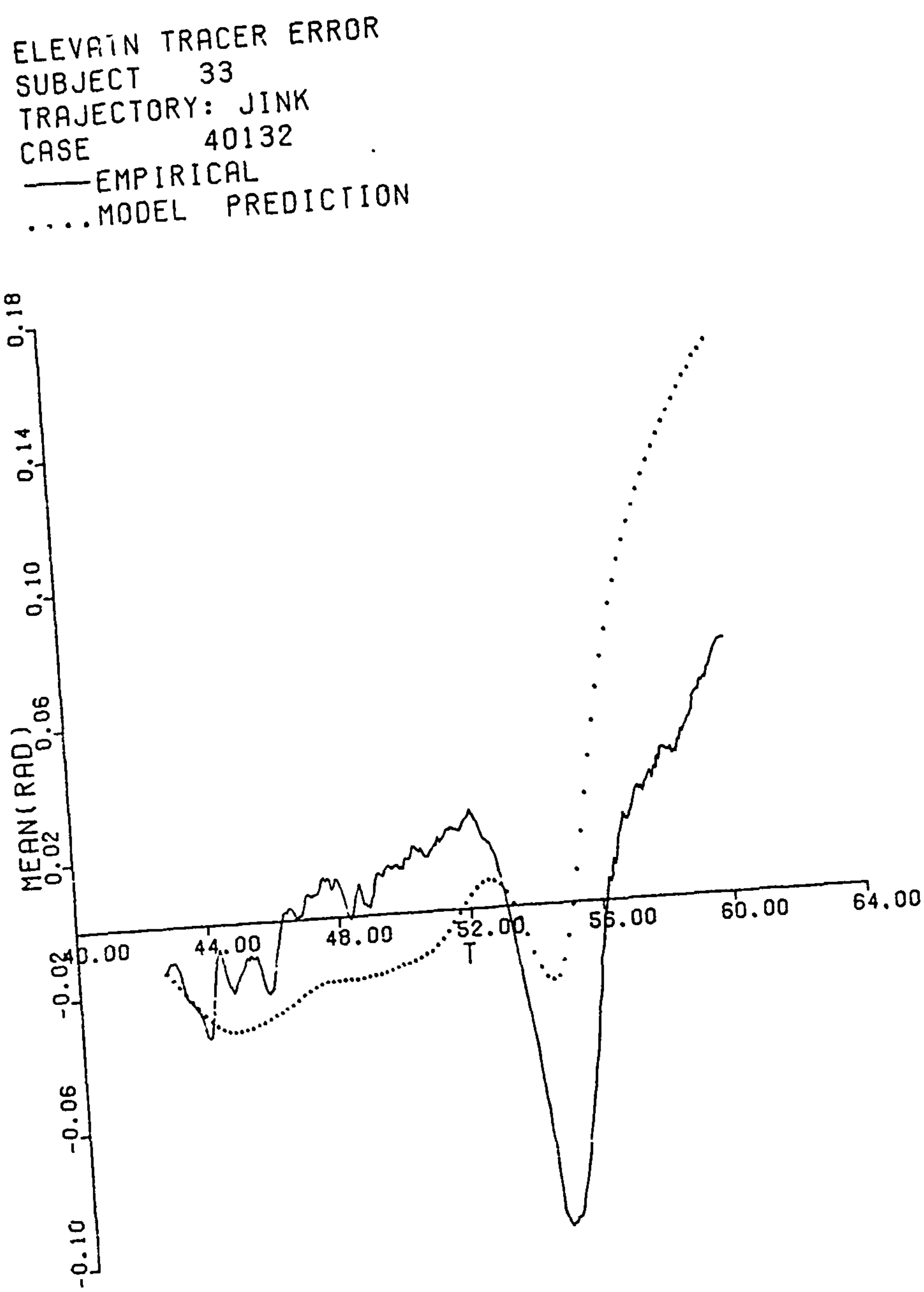


Figure 22a. Mean Tracer Error--Elevation--Jink

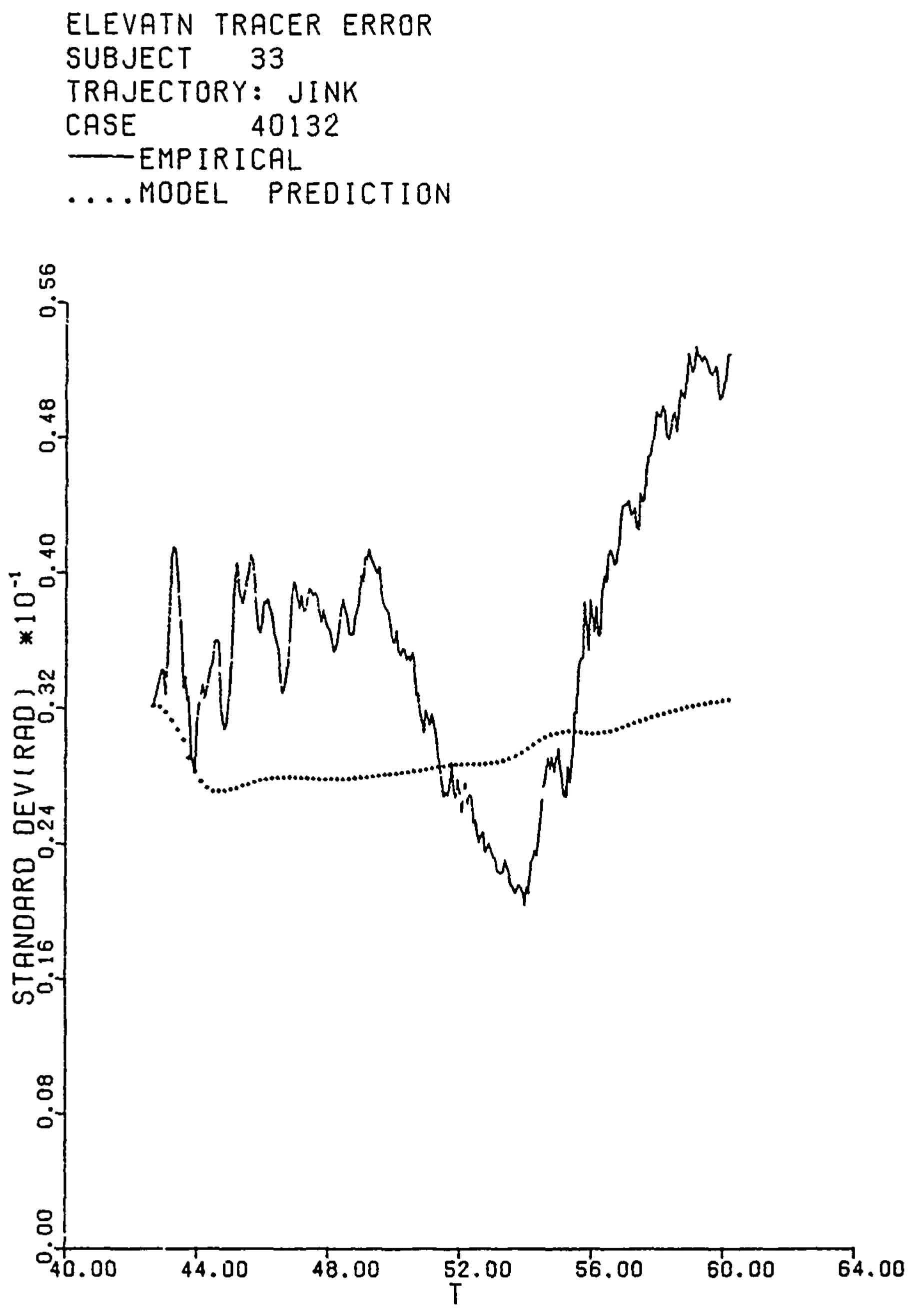


Figure 22b. Standard Deviation of Tracer Error—Elevation—Jink

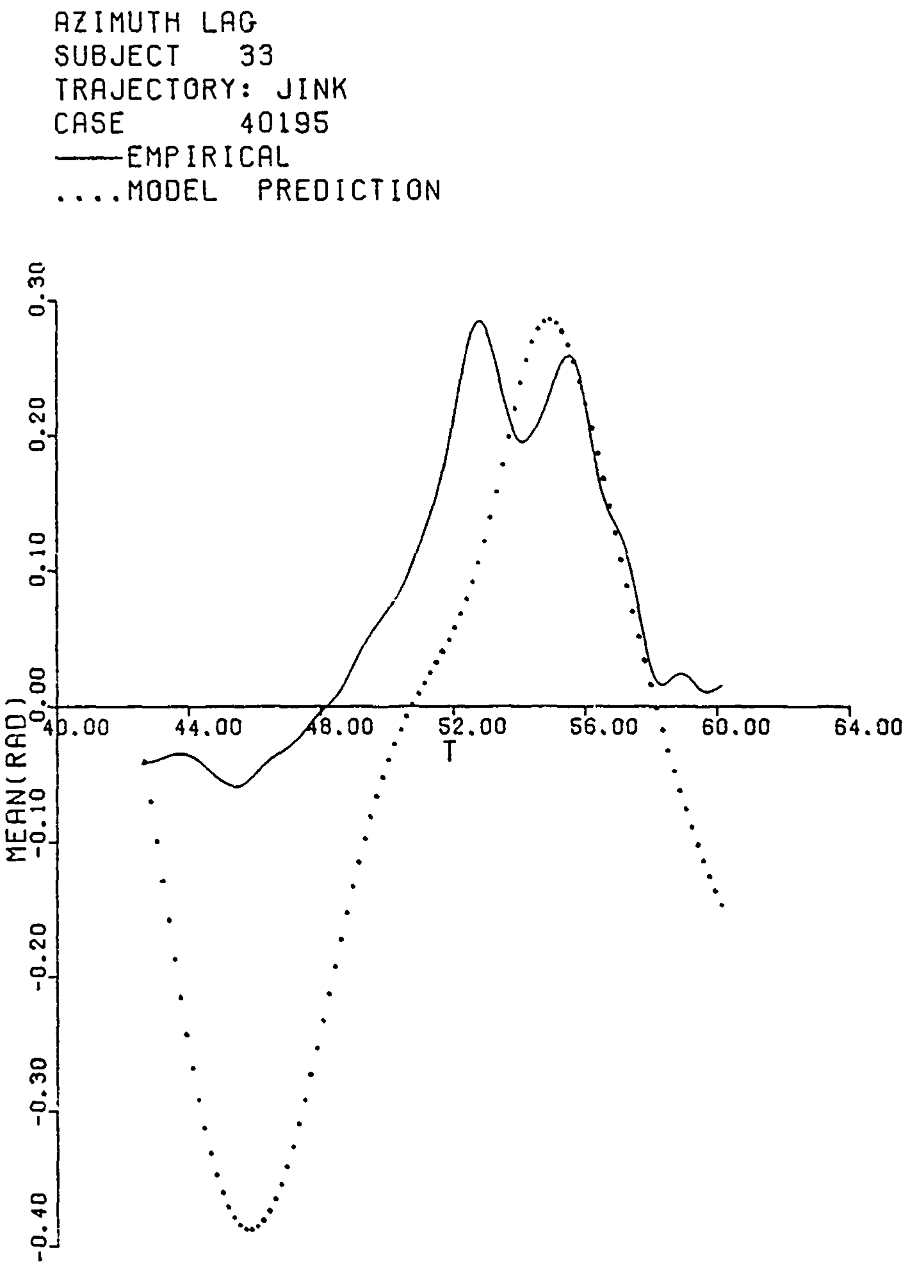


Figure 23a. Mean Tracking Error--Azimuth--Jink

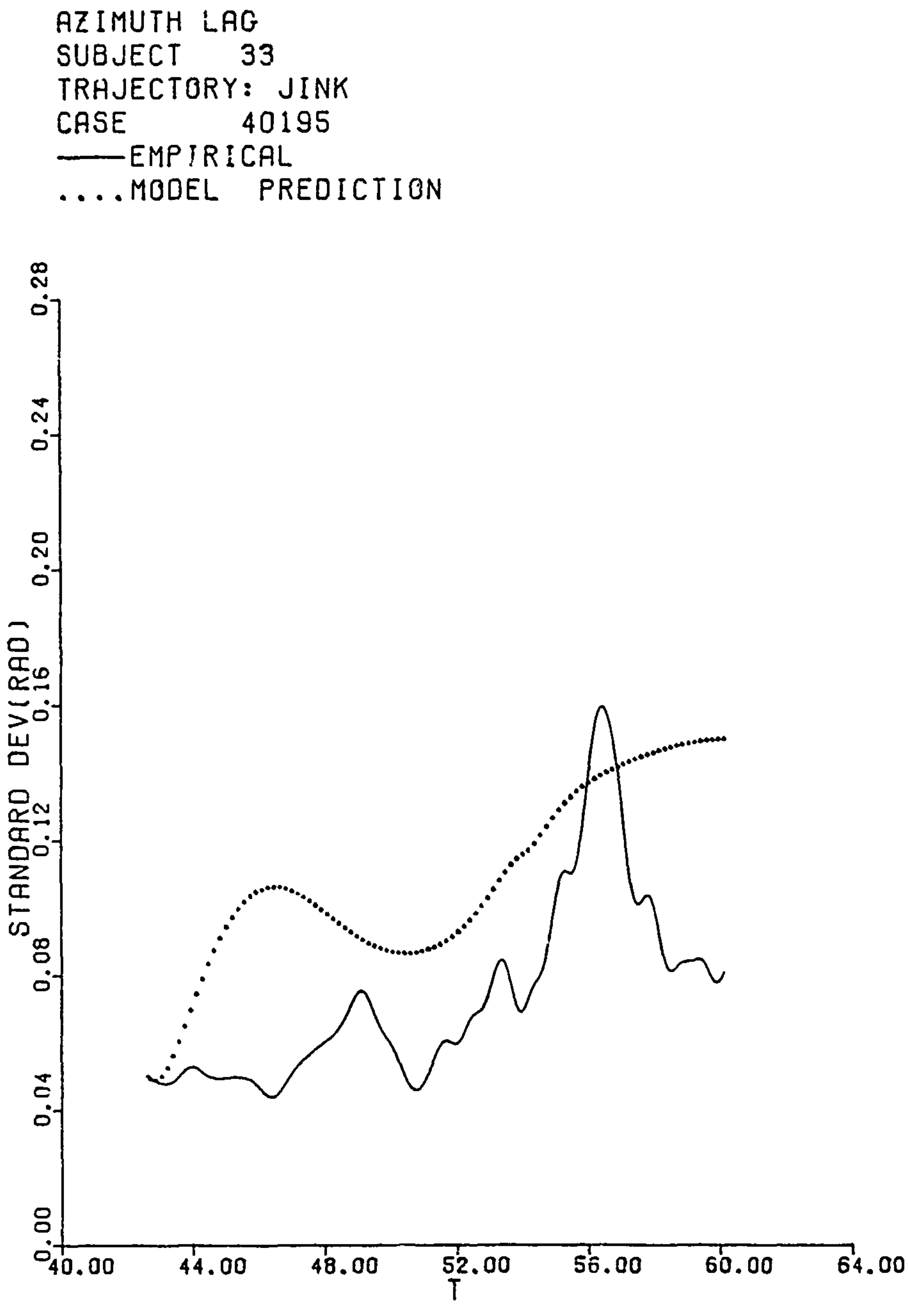


Figure 23b. Standard Deviation of Tracking Error--Azimuth--Jink

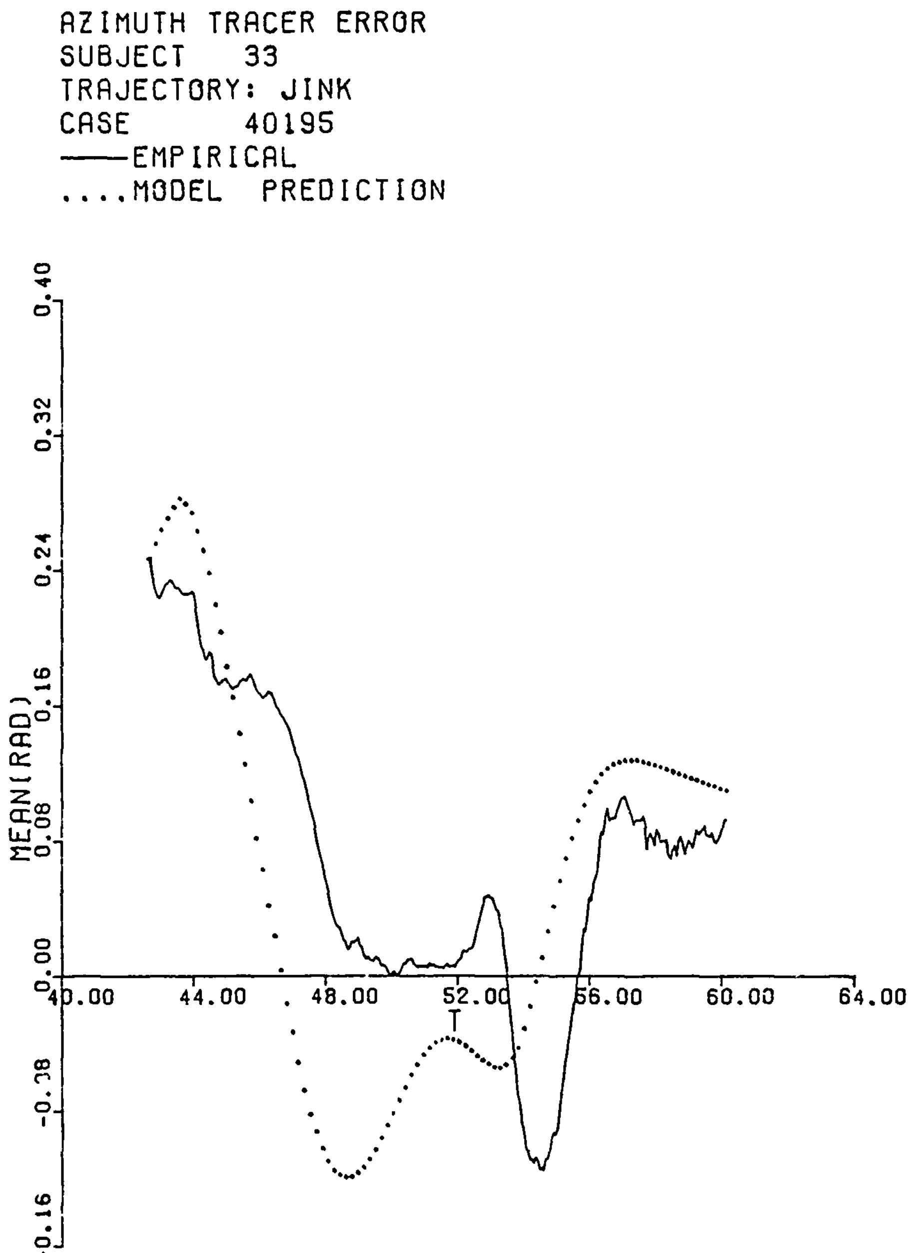


Figure 24a. Mean Tracer Error--Azimuth--Jink

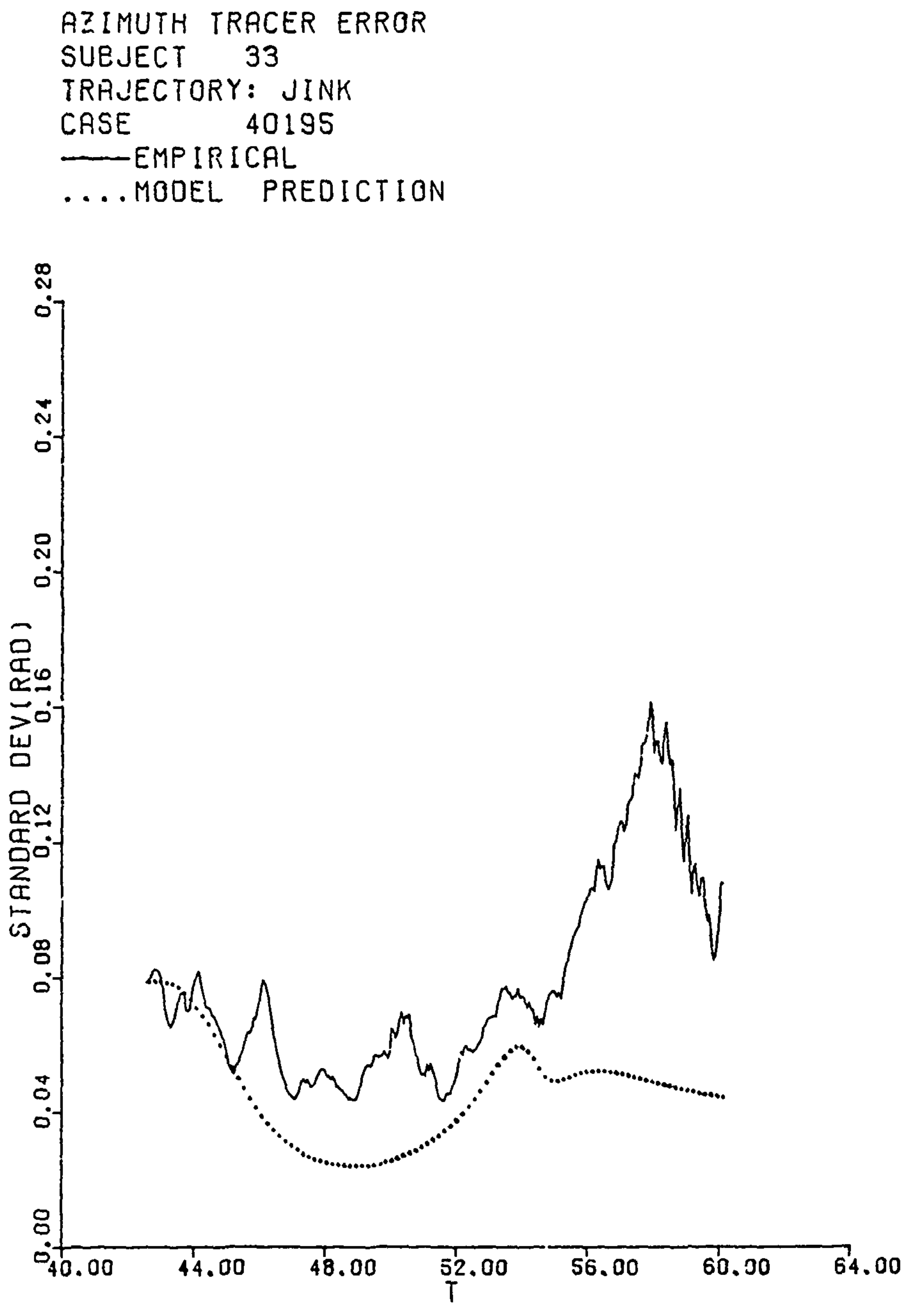


Figure 24b. Standard Deviation of Tracer Error--Azimuth--Jink

Section VI
CONCLUSION

This report summarizes the modeling of a gunner's performance in a complex AAA tracking and firing task. In this task, the gunner observes not only the tracking error, but also the miss distance of tracers from the target. A gunner model is proposed here which consists of a reduced-order observer, a linear feedback controller, and a remnant element. The Average Approximation Method is used to solve the closed-loop delay differential equations. The parameters of the model are determined via a least-squares minimization algorithm. Computer simulation results show that the model predictions of mean tracking and tracer errors are in close agreement with empirical data for several flyby and maneuvering trajectories. These results demonstrate that the proposed model is an accurate and efficient model for representing the gunner's performance characteristics in an AAA weapon system in this tracking and firing operational mode.

This human operator gunner model has been incorporated into the MTQ series of P001/OBS AAA engagement models and is designated as program P001/OBS 3/6. It is this final, composite program which is intended for use by aircraft survivability in performing their weapons effectiveness studies. Documentation of P001/OBS 3/6 is in preparation at the Air Force Aerospace Medical Research Laboratory and will be distributed separately.

APPENDIX A
PROJECTILE BALLISTICS AND INITIAL CONDITIONS

PROJECTILE BALLISTICS

The projectile ballistics equations* for an AAA weapon system used in this study can be described by

$$R(t) = \frac{v_o \cdot t}{1 + t \cdot \left\{K_1 + K_2 t - \sqrt{K_3 t^2 - K_4 t + K_5}\right\}} \tag{A1}$$

$$\alpha(t) = \sin^{-1}\left\{\frac{\cos \theta_{1B}(o) \sin \alpha_o(t)}{\left[1 + \sin^2 \alpha_o(t) - 2 \sin \alpha_o(t) \sin \theta_{1B}(o)\right]^{\frac{1}{2}}}\right\} \tag{A2}$$

$$\alpha_o(t) = 0.001\ (K_6 t + K_7 t^2) \tag{A3}$$

with constants

v_o = 930.0 m/sec
K_1 = 0.19036681/sec
K_2 = 0.14851772/sec^2
K_3 = 0.02565365/sec^4
K_4 = 0.02274266/sec^3
K_5 = 0.01660118/sec^2
K_6 = 5.202815
K_7 = 0.4863915

*See Milenski, J., "Methodology to Characterize 25 mm Projectile Trajectories for Monte Carlo Simulations," Technical Memorandum to J. Bode et al., Braddock Dunn MacDonald, November 24, 1975.

where $\theta_{1B}(o)$ is the gun barrel elevation angle at time of fire $t = o$, $r(t)$ is the range of the projectile at time t, $\alpha(t)$ is the drop of projectile elevation angle at time t. Refer to Figure 25.

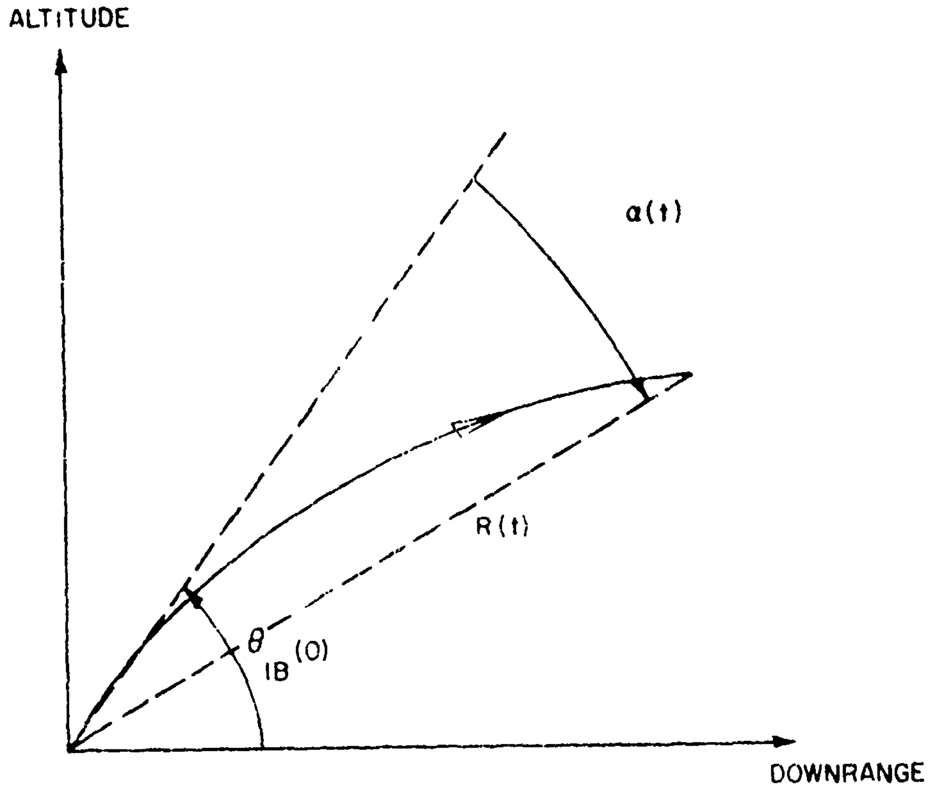


Figure 25. Schematic of Projectile Ballistics

Since the time-of-flight of interest is less than or equal to 7.5 seconds, it can be shown that $\alpha_o(t)$ is less than 0.06 rad. Equation (A2) can be approximated by

$$\alpha(t) \simeq \sin^{-1}\left[\cos\theta_{1B}(o)\cdot\alpha_o(t)\right] \qquad (A4)$$

Again, since $\cos\theta_{1B}(o) < 1$ the same argument implies $\alpha(t)$ can be approximated by

$$\alpha(t) \simeq \alpha_o(t) \cos \theta_{1B}(o)$$

$$= 0.001\ (5.202815\ t + 0.4863915\ t^2) \cos \theta_{1B}(o) \qquad \text{(A5)}$$

On the other hand, within the range of interest, it can be shown that

$$\left| K_2 t - \sqrt{K_3 t^2 - K_4 t + K_5} \right| \leq 0.028 << K_1 \qquad \text{(A6)}$$

We can approximate Equation (A1) by

$$R(t) \simeq \frac{v_o \cdot t}{1 + K_1 t} \qquad \text{(A7)}$$

Thus, for a given range R, we can solve for the time of flight from Equation (A7)

$$\tau = \frac{R}{930 - 0.19R} \qquad \text{(A8)}$$

INITIAL CONDITIONS

In the computer simulation program, the gunner starts firing at $t = t_o - \tau$ with the first tracer round reaching target's slant range at $t = t_o$. The gunner model is, therefore, turned on at $t = t_o$. The initial conditions used in the gunner model are computed as follows.

Elevation:

$$\theta_{1B}(t_o-\tau) = -\tau\,\dot{\theta}_{1T}(t_o-\tau) - 0.001\,(5.2\tau + 0.486\tau^2)$$

$$\cos\left[\theta_{1T}(t_o-\tau) + 0.05\,*\right]$$

$$\theta_{1B}(t_o) = -\hat{\tau}\,\dot{\theta}_{1T}(t_o) - 0.001\,(5.2\hat{\tau} + 0.486\hat{\tau}^2)\cos[\theta_{1T}(t_o) + 0.05]$$

$$w_1(t_o) = \theta_{1T}(t_o) - \theta_{1B}(t_o)$$

$$w_2(t_o) = \theta_{1T}(t_o) - \left\{\theta_{1T}(t_o-\tau) - \theta_{1B}(t_o-\tau) + 0.001\,(5.2\tau + 0.486\tau^2)\cos[\theta_{1T}(t_o-\tau) + 0.05)]\right\}$$

$$w_3(t_o) = w_4(t_o) = 0$$

$$P_{11}(t_o) = P_{22}(t_o) = 0.0002,\quad P_{33}(t_o) = P_{44}(t_o) = 0$$

$$P_{ij}(t_o) = 0 \qquad \forall\ i \neq j,\quad i,j = 1,2,3,4$$

*Based on empirical data, an estimate of elevation tracking error of value -0.05 rad is used here.

Azimuth:

$$\theta_{2B}(t_o-\tau) = -0.025 \times \text{sgn}\left[\dot{\theta}_{2T}(t_o-\tau)\right]$$

$$\theta_{2B}(t_o) = -\hat{\tau}\ \dot{\theta}_{2T}(t_o)$$

$$w_1\ (t_o) = \left[\theta_{2T}(t_o) - \theta_{2B}(t_o)\right] \cos \theta_{1B}(t_o)$$

$$w_2(t_o) = \left[\theta_{2T}(t_o) - \theta_{2B}(t_o-\tau)\right] \cos \theta_{1B}(t_o)$$

$$w_3(t_o) = w_4(t_o) = 0$$

$$P_{11}(t_o) = P_{22}(t_o) = 0.0002 \times \cos\ \theta_{1B}(t_o),\ P_{33}(t_o) = P_{44}(t_o) = 0$$

$$P_{ij}(t_o) = 0 \quad \forall\ i \neq j,\ i,j = 1,2,3,4$$

where $\hat{\tau}$ is the anticipated time of flight of the tracer fired at $t = t_o$. This value is normally available in the engagement model as the difference between the time of intercept and the time of fire. In general, a rough approximation can be obtained by using Equation (A8) with $R = R(t_o)$.

APPENDIX B

A HILL-CLIMBING OPTIMIZATION ALGORITHM

Let $\underline{a}$ be a vector of dimension $p \times 1$. The cost function J in Equation (30) can be rewritten as

$$\min_{\underline{a}} J(\underline{a}) = \min_{\underline{a}} \sum_{j=1}^{2} \int_{t_o}^{t_f} \left\{ \left[c^{-1} v_j(t) - \bar{\bar{x}}_j(t) \right]^2 + \ell \left[(c^{-1} s_j(t) - \bar{s}_j(t) \right]^2 \right\} dt \qquad \text{(B1)}$$

The iteration procedures which adjust the values of $\underline{a}$ in search for a minimum of $J(\underline{a})$ are described below.

1. Choose an initial guess $\underline{a}_o$ of $\underline{a}$, and ε which is a termination lower bound. If the absolute sum of changes in $\underline{a}$ within one iteration is less than or equal to ε, the iteration process will be terminated.

2. Compute $J(\underline{a}_o)$ and call it J_o.

3. (a) Choose a set of p orthornormal vectors $\underline{v}_i$. cumulative vector $\underline{\xi}_i$, increment coefficients e_i, cumulative increment coefficients d_i, success indices q_i, $i = 1,2, \ldots, p$.

   Initially, let

   $$\underline{v}_1 = \underline{\xi}_1 = (1,0, \ldots, 0)^T$$

   $$\underline{v}_2 = \underline{\xi}_2 = (0,1,0, \ldots, 0)^T$$

   $$\vdots$$

$$\underline{v}_p = \underline{\xi}_p = (0, \ldots, 0,1)^T$$

$$e_i = 0.1$$

$$d_i = 0$$

$$q_i = 2$$

$$i = 1,2, \ldots, p$$

(b) Set parameter axis loop index $i = 1$

4. Adjust the parameter vector according to

$$\underline{a}_{new} = \underline{a}_{old} + e_i \underline{v}_i$$

5. Check constraints on $\underline{a}$, e.g., if $\underline{a}(i)$ are constrained to be nonnegativ then set

$$\underline{a}_{new}(i) = \left|\underline{a}_{new}(i)\right|, \text{ if } \underline{a}_{new}(i) < 0$$

6. Compute $J(\underline{a}_{new})$ and call it J_1. Then, do the comparison test. If $-J_1 \geq -J_o$, set $J_o = J_1$, proceed to (7); otherwise, proceed to (9).

7. (a) Store accumulated increment along each axis

$$d_i = d_i + e_i$$

(b) Set $\underline{a}_{old} = \underline{a}_{new}$

(c) Reset increment unit for next subiteration

$$e_i = 3e_i$$

8. Test the success index q_i. If $q_i > 1.5$, set $q_i = 1$. Proceed to (11).

9. Reverse climbing direction and cut the increment length by half

$$e_i = -\frac{1}{2} e_i$$

10. Test the success index q_i. If $q_i \leq 1.5$, set $q_i = 0$.

11. Test all success indices q_j. If $q_j < 0.5$ for all $j = 1,2, \ldots,$ p, compute

$$\| \underline{\xi}_1 \|, \frac{\| \underline{\xi}_2 \|}{\| \underline{\xi}_1 \|}$$

where

$$\| \underline{\xi}_j \| = \left[\sum_{k=1}^{p} \underline{\xi}_j^2(k) \right]^{\frac{1}{2}}$$

12. Test the parameter loop index i. If $i = p$, proceed to (13). Otherwise, set $i = i + 1$, proceed to (4).

13. Test the termination bound. If

$$\sum_{i=1}^{p} \left| d_i \right| \leq \varepsilon$$

stop the iteration; otherwise, proceed to (14).

14. Rotate orthormal axes $\underline{v}_i$ according to Gram-Schmidt orthogonalization procedure. First, store cumulative vecor $\underline{\xi}_i$

$$\underline{\xi}_1 = d_1\underline{v}_1 + d_2\underline{v}_2 + \ldots + d_p\underline{v}_p$$

$$\underline{\xi}_2 = \qquad d_2\underline{v}_2 + \ldots + d_p\underline{v}_p$$

$$\vdots$$

$$\underline{\xi}_p = \qquad\qquad d_p\underline{v}_p$$

Second, compute a set of orthogonal changes β_i, and new axes $\underline{v}_i$

$$\begin{cases} \beta_1 = \underline{\xi}_1 \\ \underline{v}_1 = \dfrac{\underline{\beta}_1}{\| \underline{\beta}_1 \|} \end{cases}$$

$$\begin{cases} \underline{\beta}_2 = \underline{\xi}_2 - \left(\underline{\xi}_2^T \, \underline{v}_1\right)\underline{v}_1 \\ \underline{v}_2 = \dfrac{\underline{\beta}_2}{\| \underline{\beta}_2 \|} \end{cases}$$

$$\vdots$$

$$\begin{cases} \underline{\beta}_p = \underline{\xi}_p - \displaystyle\sum_{i=1}^{p-1} \underline{\xi}_p{}^T \underline{v}_i \ \underline{v}_i \\ \\ \underline{v}_p = \dfrac{\underline{\beta}_p}{|| \underline{\beta}_p ||} \end{cases}$$

where

$$|| \underline{\beta}_j || = \left[\sum_{i=1}^{p} \beta_j{}^2(i) \right]^{\frac{1}{2}}$$

$$\underline{\xi}_j{}^T \underline{v}_k = \sum_{i=1}^{p} \xi_j(i) \ v_k(i)$$

Proceed to 3(a) with new $\underline{v}_i$ but retain $e_i = 0.1$, $d_i = 0$, $q_i = 2$.

APPENDIX C

LISTING OF PARAMETER IDENTIFICATION PROGRAM--

ELEVATION CASE

```
MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM

W6W,T350,CM79300,  L760295,WEI,256-3960
MAP(ON).
COMMENT.   **TIME-VARYING T.O.F.
COMMENT.   **ELFIT,ID=L760295,CY=3; THETPDOT=ALPHA(1)*THETB(T-TAU)**
FTN.
ATTACH,TAPE1,DATM6RAZ,CY=3,MR=1.
ATTACH,TAPE2,DATM6TRACER,CY=3,MR=1.
LGO.
      PROGRAM OPT(TAPE1,TAPE2,INPUT,OUTPUT)
      DIMENSION ALPHA(8),PSI(8),A(8),D(8),ALPHAN(8)
      DIMENSION SIMP(7),E(8),V(8,8),Z(8,8),XI(8,8),G(8,8),F(7),F2(7,8)
      COMMON/ARRAY/X(1000),S(1000),ELDD(1000),TX(1000),TS(1000),RAN(1000
     1)
      COMMON/S/TAU,C0,DEL,NSTP,NDIM,T0,IPT,Y10
      LOGICAL FM, AF
      INTEGER Q,SUBIT,FR
      LG=0
      DO 1 I=1,8
      DO 1 J=1,8
      V(I,J)=XI(I,J)=0.0
      IF (I.EQ.J) V(I,J)=XI(I,J)=1.0
      Z(I,J)=0.0
      G(I,J)=0.0
1     CONTINUE
      FR=1
      READ*,K1,NDIM,T0,IPT,Y10
      PRINT 42,K1,NDIM,T0,IPT,Y10
42    FORMAT(1H1,"NO.OF PTS = ",I4,2X,"ORDER= ",I2,
     C 2X,"INIT TIME= ",G12.5//1X,"READ EVERY ",I2," POINT","Y10= ",G12
     C.5)
      KT=T0/0.03
      K=K1-KT
      READ(1,43)(DUM6,AZ,AZD,AZDD,EL,ELD,DUM1,T,AZMN,DUM2,AZSD,DUM3,
     C I=1,KT)
      DO 44 I=1,K
      READ(1,43) DUM6,AZ,AZD,AZDD,EL,ELD,DUM1,T,AZMN,DUM2,AZSD,DUM3
      IF(EOF(1)) 45,45
43    FORMAT(12G11.5)
45    IF(MOD(I-1,IPT).NE.0) GO TO 44
      IH=(I-1)/IPT+1
      ELDD(IH)=DUM1
      X(IH)=DUM2
      S(IH)=DUM3
      RAN(IH)=7.5
      IF(DUM6.LT.2077.) RAN(IH)=DUM6/(930.-.19*DUM6)
44    CONTINUE
      DO 47 I=1,K
```

```
      READ(2,46)TIME,TAZ,DUM4,TAZSD,DUM5
      IF(EOF(2))46,48
46    FORMAT(5G12.5)
48    IF(MOD(I-1,IPT).NE.0) GO TO 47
      IG=(I-1)/IPT+1
      TX(IG)=DUM4
      TS(IG)=DUM5
47    CONTINUE
      DEL=0.03*IPT
      EG=1.34
      NSTP=IH-1
      READ*,(ALPHA(I),I=1,8)
      READ*,EPS
      AJ=0.0
10    DO 8 I=1,8
      E(I)=.1
      D(I)=0.0
      A(I)=2.0
8     CONTINUE
      IT=SUBIT=0
      CALL INTG(ALPHA,AJ)
      OLDJ=AJ
      PRINT 40,ALPHA
40    FORMAT("0ALPHA= ",G12.5,"*",G12.5,"*",G12.5,"*",G12.5,"*",
     C G12.5,"*",G12.5,"*",G12.5,"*",G12.5)
      PRINT 41,AJ,IT,SUBIT
41    FORMAT(" J= ",G12.5," ITERATIONS= ",I5,"SUBITERATIONS= ",I5)
11    DO 130 K=1,8
      SUBIT=SUBIT+1
      DO 12 L=1,8
      ALPHAN(L)=ALPHA(L)+E(K)*V(K,L)
      IF((L.GE.3).AND.(L.LE.5)) GO TO 12
      IF (ALPHAN(L).LT.0.0) ALPHAN(L)=-ALPHAN(L)
12    CONTINUE
      CALL INTG(ALPHAN,AJ)
      IF (AJ.GT.OLDJ) GO TO 20
      OLDJ=AJ
      PRINT 40,ALPHAN
      PRINT 41,AJ,IT,SUBIT
      D(K)=D(K)+E(K)
      E(K)=3*E(K)
      DO 15 M=1,8
      ALPHA(M)=ALPHAN(M)
15    CONTINUE
      IF (A(K).GT.1.5) A(K)=1.0
      GO TO 25
20    E(K)=-.5*E(K)
      IF (A(K).LE.1.5) A(K)=0.0
25    CK=0.0
      DO 30 L=1,8
      IF (A(L).LT.0.5)GO TO 30
      CK=1.0
```

```
30    CONTINUE
      ~~IF (GK.NE.0.0)GO TO 100~~
      SUM1=SUM2=0.0
      ~~DO 32 M=1,8~~
      SUM1=SUM1+XI(1,M)**2
      ~~SUM2=SUM2+XI(2,M)**2~~
32    CONTINUE
      ~~X1=SQRT(SUM1)~~
      X2=SQRT(SUM2)
      ~~X3=X1/X2~~
      PRINT 33,OLDJ,ALPHA,X1,X3
33    ~~FORMAT (*1J*      ,G12.5/* ALPHA*(*,G12.5,*,*,G12.5,*,*,G12.5,~~
     2*,*,G12.5,*,*,G12.5,*,*,G12.5,*,*,G12.5,*,*,G12.5,*)*/
     ~~3* XI(1)=*,G12.5/* XI(1)/XI(2)=*,G12.5)~~
      GO TO 110
100   ~~CONTINUE~~
105   GO TO 11
110   ~~SUM3=0.0~~
      DO 115 L=1,7
      ~~DO 115 M=1,8~~
      F(L)=0.0
      ~~F2(L,M)=0.0~~
115   CONTINUE
      ~~DO 117 J=1,8~~
      DO 117 K=1,8
117   ~~XI(J,K)=0.0~~
      DO 120 I=1,8
120   ~~SUM3=SUM3+(ABS(D(I)))~~
      IF (SUM3.LE.EPS) GO TO 1000
      DO 130 N=1,8
      DO 130 J=1,8
      ~~Z(N,J)=D(N)*V(N,J)~~
130   CONTINUE
      ~~DO 140 J=1,8~~
      DO 140 L=1,
      ~~DO 140 K=J,8~~
      XI(J,L)=XI(J,L)+Z(K,L)
140   ~~CONTINUE~~
      SUM4=0.0
      ~~DO 150 J=1,8~~
150   SUM4=SUM4+XI(1,J)**2
      ~~SUM4=SQRT(SUM4)~~
      DO 155 J=1,8
155   ~~V(1,J)=XI(1,J)/SUM4~~
      KOUNT=2
159   ~~Q=KOUNT-1~~
      DO 175 K=1,Q
      ~~DO 160 L=1,8~~
      F(Q)=F(Q) + XI(KOUNT,L)*V(K,L)
160   CONTINUE
      DO 170 M=1,8
      F2(Q,M)=F(Q)*V(K,M) + F2(Q,M)
170   CONTINUE
      ~~F(Q)=0.0~~
```

```
175     CONTINUE
        DO 190 I=1,8
190     G(KOUNT,I)=XI(KOUNT,I)-F2(Q,I)
        SUM5(Q)=0.0
        DO 200 M=1,8
200     SUM5(Q)=SUM5(Q)+G(KOUNT,M)**2
        SUM5(Q)=SQRT(SUM5(Q))
        DO 215 M=1,8
        V(KOUNT,M)= (KOUNT,M)/SUM5(Q)
215     CONTINUE
        KOUNT=KOUNT+1
        IF (KOUNT.LE.8) GO TO 159
        IT=IT+1
        SUBIT=0
        FR=1
        DO 250 K=1,8
        E(K)=.1
        G(K)=...
        A(K)=2.0
250     CONTINUE
        GO TO 11
1000    CALL EXIT
        END
        SUBROUTINE INTG(ALPHA,EJ)
        COMMON/ARRAY/XEMP(1000),SEMP(1000),EDD(1000),TX(1000),TS(1000)
       1,TAU(1000)
        COMMON/B/STAU,CO,DEL,NSTP,NO,TO,IPT,Y10
        DIMENSION W(4),F(4,4),P1(4,4),P2(4,4),ALPHA(8),A(4,4)
       1,B(16),EB(16),EBINT(16),EA(4,4),EAINT(4,4),F(4),COC(4,4),D(4)
       2,X3(1000),X4(1000),EDH(1000),EEDH(1000)
        EQUIVALENCE (A(1,1),B(1)),(EA(1,1),EB(1)),(EAINT(1,1),EBINT(1))
        DATA WT/1./
C
C--INITIALIZATION
C
        SGAL=G)**2/DEL
        DO 1 I=1,ND
        DO 1 J=1,ND
        P(I,J)=0.
        COC(I,J)=0.
1       A(I,J)=0.
        NC1=ND-1
        NC2=ND-2
        NAA=NC/2-1
        DO 11 I=1,ND
        W(I)=0.
        D(I)=0.
        F(I)=0.
11      P(I,I)=0.00 0
        P(1,1)=0.00 22437
        P(2,2)=0.00 053246
        X3(1)=0.000436
        X4(1)=0.
        EDH(1)=X3(1)
        EDDH(1)=0.
        S1=0.
        ARG=-DEL*ALPHA(2)
        IF(ARG.GT.-20.) S1=EXP(ARG)
```

```
C
C-COMPUTE AND STORE STATES X3 AND X4
C
C     PRINT 97
97    FORMAT(4X,"TIME",4X,"TARGET VEL",2X,"EST VEL ERROR",2X,"EST TAR VE
     1 L"/)
      T=T0
      DO 10 KK=1,NSTP
      T=T+DEL
C     IF((MOD(KK,100).EQ.0).OR.(KK.EQ.1)) PRINT 96,T,X3(KK),X4(KK)
C    1 ,X3(KK)-X4(KK)
96    FORMAT(4G12.5)
      K1=KK+1
      K2=KK-1
      X3(K1)=X3(KK)+EDD(KK)*DEL
      IF(ALPHA(2).EQ.0.)GO TO 4
      X4(K1)=S1 X4(KK)+EDD(KK)*(1.-S1)/ALPHA(2)
      GO TO 3
4     X4(K1)=X4(KK)+EDD(KK)*DEL
3     CONTINUE
C
C-COMPUTE AND STORE ESTIMATED TARGET VELOCITY AND ACCELERATION
C
      EDH(KK)=X3(KK)-X4(KK)
      IF(K2.GE.1) EDDH(KK)=(EDH(KK)-EDH(K2))/DEL
10    CONTINUE
      NDIM=ND
      N1=ND**2
C
C START INTEGRATION LOOP
C
      KP=1
      T=T0
      W(1)=Y13
      W(2)=0.024786
      SMEAN=(W(1)-XEMP(125))**2+(W(2)-TX(1))**2
      SSD=(SQRT(P(1,1))-SEMP(125))**2+(SQRT(P(2,2))-TS(1))**2
      DO 100 KK=125,NSTP
      K1=KK+1
      M=TAU(KK)/DEL
      RA=NAA/TAU(KK)
      A2=1.-(TAU(K1)-TAU(KK))/DEL
      CGR=CC*ALPHA(3)
      A(1,1)=-CGR
      A(1,2)=-CC*ALPHA(4)
      A(2,ND1)=A(1,1)-ALPHA(1)*A2
      A(2,ND)=A(1,2)*ALPHA(1)*A2
      DO 2 I=1,ND2
      J1=I+2
      A(J1,I)=RA
      A(J1,J1)=-RA
2     CONTINUE
      CALL DSORT(NDIM,E,DEL,FB,EBINT,5)
      CR3=CC*ALPHA(5)-ALPHA(1)*A2
      CR4=CC*ALPHA(5)
```

```
      SCAL1=SCAL+(ALPHA(1)*A2)**2
      GGC(2,2)=ALPHA(6)*SCAL1
C     IF((MOD(KK,10).EQ.0).OR.(KK.EQ.1)) PRINT 99,T,W(1)+Y1),SMEAN,
C    1 SQRT(P(1,1)),SSC
99    FORMAT(5G12.5)
C
C COMPUTE MEAN TRACKING ERROR
C
      DO 110 I=1,NDIM
      DO 120 J=1,NDIM
      D(I)=D(I)+EA(I,J)*W(J)
120   CONTINUE
110   CONTINUE
      F(1)=(1.-CR4) X3(KK)+CR4* X4(KK)
      F(2)=X3(KK)
      IF(KK.GT.M)  F(2)=F(2)+CR3*(X4(KK-M)-X3(KK-M))
      DO 130 I=1,NDIM
      DO 140 J=1,
      D(I)=D(I)+EAINT(I,J)*F(J)
140   CONTINUE
      W(I)=D(I)
      D(I)=0.
130   CONTINUE
C
C COMPUTE ERROR DUE TO MEAN TRACKING ERROR
C
      SMEAN=SMEAN+(W(1)-XEMP(K1))**2+(W(2)-TX(K1-125))**2
C
C COMPUTE COVARIANCE MATRIX
C
      GGC(1,1)=(ALPHA(6)+ALPHA(7)*ABS(EDH(KK))+ALPHA(8)*ABS(EDDH(KK)))
     1 *SCAL
      IF(KK.GT.M) GGC(2,2)=(ALPHA(6)+ALPHA(7)*ABS(EDH(KK-M))
     1   +ALPHA(8) ABS(EDDH(KK)))*SCAL1
      CALL MULT(EAINT,GGC,NDIM,N1,P1,10)
      CALL MULT(EA,P,NDIM,N1,P2,10)
      DO 220 I=1,NDIM
      DO 220 J=1,NDIM
      P(I,J)=P1(I,J)+P2
220   CONTINUE
C
C COMPUTE ERROR DUE TO STANDARD DEVIATION
C
      SSD=SSD+(SQRT(P(1,1))-SEMP(K1))**2+(SQRT(P(2,2))-TS(K1-125))**2
      T=T+DEL
      KP=KP+1
100   CONTINUE
      EJ=DEL*(SMEAN+WT*SSD)
      RETURN
      END
      SUBROUTINE MULT(E,F,L,L1,H,MR)
      DIMENSION E(1),F(1),G(16),H(1)
      DO 10 I=1,L
```

```
      II=1
      DO 10 K=1,L
      TEMP=C.
      DO 5 J=I,L1,L
      TEMP=TEMP+E(J)*E(II)
5     II=II+1
      KK=(K-1)*L+I
      H(KK)=TEMP
10    G(KK)=TEMP
      IF(MR.EQ.1)RETURN
      DO 20 I=1,L
      DO 20 K=I,L
      TEMP=C.
      II=K
      DO 15 J=I,L1,L
      TEMP=TEMP+G(J)*E(II)
15    II=II+L
      KK=(K-1)*L+I
20    H(KK)=TEMP
      L2=L-1
      DC 30 I=1,L2
      L3=I+1
      DO 30 J=L3,L
      K1=(I-1)*L+
      K2=(J-1)*L+I
30    H(K1)=H(K2)
      END
      SUBROUTINE ESGRT(NDIM,A,DEL,EA,EAINT,NT)
      DIMENSION A(1),EA(1),EAINT(1),COEF(30)
C     SETS EA=EXP(A*DEL),EAINT-INTEGRAL EA 0 TO DEL
      NDIM1=NDIM+1
      NN=NDIM*NDIM
      NTM1=NT-1
      COEF(NT)=1.
      DO 10 I=1,NTM1
      II=NT-I
   10 COEF(II)=DEL*COEF(II+1)/FLOAT(I)
C     NT MUST BE AT LEAST 3
      CALL DIAG(NDIM,EAINT,A,COEF(1),COEF(2))
      DO 60 L=3,NT
      CALL MULT(A,EAINT,NDIM,NN,EA,1)
      IF(L.EQ.NT)GO TO 70
   60 CALL DIAG(NDIM,EAINT,EA,1.0,COEF(L))
   70 DO 80 II=1,NN,NDIM1
      EA(II)=EA(II)+1.0
   80 CONTINUE
      END
```

```
      SUBROUTINE DIAG(NDIM,A,B,C1,C2)
      DIMENSION A(1),B(1)
      NDIM1=NDIM+1
      NN=NDIM*NDIM
      NM1=NDIM-1
      II=1
      IF(C1.EQ.1.0) GO TO 10
      DO 5 J=1,NN,NDIM
      K=J+NM1
      DO 4 I=J,K
    4 A(I)=C1*B(I)
      A(II)=A(II)+C2
    5 II=II+NDIM1
      RETURN
   10 DO 7 J=1,NN,NDIM
      K=J+NM1
      DO 6 I=J,K
    6 A(I)=B(I)
      A(II)=A(II)+C2
    7 II=II+NDIM1
      RETURN
      END
2208,4,30.6,2,-6.,.434
1.1,2.,1.5577,1.6419,.92133,.11526E-4,.14415E-4,.46165E-3
0.01
```

APPENDIX D

LISTING OF PARAMETER IDENTIFICATION PROGRAM--

AZIMUTH CASE

```
HBW,T300,CM75000.  L760295,WEI,256-3960
MAP(ON).
COMMENT.      *NO DECK AF
COMMENT. **AZFIT,ID=L760295,CY=3; IDENTIFY 7 PARAMETERS**
FIN.
ATTACH,TAPE1,DATM6RAZ,CY=3,MR=1.
ATTACH,TAPE2,DATM6TRACER,CY=3,ID=L760295,MR=1.
LGO.
      PROGRAM OPT(TAPE1,TAPE2,INPUT,OUTPUT)
      DIMENSION ALPHA(8),PSI(8),A(8),D(8),ALPHAN(8)
      DIMENSION SUMS(7),E(8),V(8,8),Z(8,8),XI(8,8),B(8,8),F(7),F2(7,8)
      COMMON/ARRAY/ X(1000),S(1000),AZDD(1000),ELS(1000),RAN(1000)
     1,TX(1000),TS(1000)
      COMMON/S/TAU,C0,DEL,NSTP,NDIM,T0,IPT,Y10
      LOGICAL FM,PAR
      INTEGER Q,SUBIT,FR
      LG=0
      NPAR=7
      NPAR1=NPAR-1
      DO 1 I=1,NPAR
      DO 1 J=1,NPAR
      V(I,J)=XI(I,J)=0.0
      IF (I.EQ.J) V(I,J)=XI(I,J)=1.0
      Z(I,J)=0.0
      B(I,J)=0.0
1     CONTINUE
      FR=1
      READ*,K1,NDIM,T0,IPT,Y10
      PRINT 42,K1,NDIM,T0,IPT,Y10
42    FORMAT(1H1,"NO.OF PTS = ",I4,2X,"ORDER= ",I2,
     C 2X,"INIT TIME= ",G12.5//1X,"READ EVERY ",I2," POINT",",Y10= ",G12
     C.5)
      KT=T0/0.03
      K=K1-KT
      READ(1,43)(DUM4,AZ,AZD,DUM1,EL,ELD,ELDD,T,DUM2,ELMN,DUM3,ELSD,
     C I=1,KT)
      DO 44 I=1,K
      READ(1,43) DUM4,AZ,AZD,DUM1,EL,ELD,ELDD,T,DUM2,ELMN,DUM3,ELSD
      IF(EOF(1)) 45,45
43    FORMAT(12G11.5)
45    IF(MOD(I-1,IPT).NE.0) GO TO 44
      IH=(I-1)/IPT+1
      AZDD(IH)=DUM1
      X(IH)=DUM2
      ELS(IH)=EL-ELMN
      S(IH)=DUM3
      RAN(IH)=7.5
      IF(DUM4.LT.2877.) RAN(IH)=DUM4/(930.-.19*DUM4)
44    CONTINUE
      DO 47 I=1,K
      READ(2,46) TIME,DUM5,TEL,DUM7,TELSD
      IF(EOF(2)) 48,48
46    FORMAT(5G12.5)
```

```
46     IF(MOD(I-1,IPT).NE.0.) GO TO 47
       IG=(I-1)/IPT+1
       TX(IG)=DUM5
       TS(IG)=DUM7
47     CONTINUE
       DEL=0.03*IPT
       CG=1.28
       NSTP=IH-1
       READ*,(ALPHA(I),I=1,NPAR)
       READ*,EPS
       PRINT ,EPS,(ALPHA(I),I=1,NPAR)
       AJ=0.0
10     DO 8 I=1,NPAR
       E(I)=.1
       D(I)=0.0
       A(I)=2.0
8      CONTINUE
       II=SUBIT=0
       CALL INTG(ALPHA,AJ)
       OLDJ=AJ
       PRINT 40,(ALPHA(MP),MP=1,NPAR)
40     FORMAT("0ALPHA= ",G12.5,"!",G12.5,"!",G12.5,"!",G12.5,"!",
     C G12.5,"!",G12.5,"!",G12.5)
       PRINT 41,AJ,II,SUBIT
41     FORMAT(" J= ",G12.5," ITERATIONS= ",I5,"SUBITERATIONS= ",I5)
11     DO 100 K=1,NPAR
       SUBIT=SUBIT+1
       DO 12 L=1,NPAR
       ALPHAN(L)=ALPHA(L)+E(K)*V(K,L)
       IF((L.GE.2).AND.(L.LE.4)) GO TO 12
       IF (ALPHAN(L).LT.0.0) ALPHAN(L)=-ALPHAN(L)
12     CONTINUE
       CALL INTG(ALPHAN,AJ)
       IF (AJ.GT.OLDJ) GO TO 20
       OLDJ=AJ
       PRINT 40,(ALPHAN(MP),MP=1,NPAR)
       PRINT 41,AJ,II,SUBIT
       D(K)=D(K)+E(K)
       E(K)=3*E(K)
       DO 15 M=1,NPAR
       ALPHA(M)=ALPHAN(M)
15     CONTINUE
       IF (A(K).GT.1.5) A(K)=1.0
       GO TO 25
20     E(K)=-.5*E(K)
       IF (A(K).LE.1.5) A(K)=2.0
25     CK=0.0
       DO 30 L=1,NPAR
       IF (A(L).LT.0.5)GO TO 30
       CK=1.0
30     CONTINUE
       IF (CK.NE.0.0)GO TO 100
       SUM1=SUM2=0.0
       DO 32 M=1,NPAR
       SUM1=SUM1+XI(1,M)**2
       SUM2=SUM2+XI(2,M)**2
```

```
32    CONTINUE
      X1=SQRT(SUM1)
      X2=SQRT(SUM )
      X3=X1/X2
      PRINT 33,OLDJ,(ALPHA(MP),MP=1,NPAR),X1,X3
33    FORMAT (*1J=    *,G12.5/* ALPHA=(*,G12.5,*,*,G12.5,*,*,G12.5,
     2*,*,G12.5,*,*,G12.5,*,*,G12.5,*,*,G12.5,*)*/
     3* XI(1)=  *,G12.5/* XI(1)/XI(2)= *,G12.5)
      GO TO 110
100   CONTINUE
105   GO TO 11
110   SUM3=0.0
      DO 115 L=1,NPAR1
      DO 115 M=1,NPAR
      F(L)=0.0
      F2(L,M)=0.0
115   CONTINUE
      DO 117 J=1,NPAR
      DO 117 K=1,NPAR
117   XI(J,K)=0.0
      DO 120 I=1,NPAR
120   SUM3=SUM3+(ABS(D(I)))
      IF (SUM3.LE.EPS)GO TO 1000
      DO 130 N=1,NPAR
      DO 130 J=1,NPAR
      Z(N,J)=D(N)*V(N,J)
130   CONTINUE
      DO 140 J=1,NPAR
      DO 140 L=1,NPAR
      DO 140 K=J,NPAR
      XI(J,L)=XI(J,L)+Z(K,L)
140   CONTINUE
      SUM4=0.0
      DO 150 J=1,NPAR
150   SUM4=SUM4+XI(1,J)**2
      SUM4=SQRT(SUM4)
      DO 155 J=1,NPAR
155   V(1,J)=XI(1,J)/SUM4
      KOUNT=2
159   Q=KOUNT-1
      DO 175 K=1,Q
      DO 160 L=1,NPAR
      F(Q)=F(Q) + XI(KOUNT,L)*V(K,L)
160   CONTINUE
      DO 170 M=1,NPAR
      F2(Q,M)=F(Q)*V(K,M) + F2(Q,M)
170   CONTINUE
      F(Q)=0.0
175   CONTINUE
      DO 190 I=1,NPAR
190   U(KOUNT,I)= I(KOUNT,I)-F2(Q,I)
      SUM5(Q)=0.0
      DO 200 M=1,NPAR
200   SUM5(Q)=SUM5(Q)+U(KOUNT,M)**2
      SUM5(Q)=SQRT(SUM5(Q))
      DO 215 M=1,NPAR
      V(KOUNT,M)=U(KOUNT,M)/SUM5(Q)
```


/ / /

```
215   CONTINUE
      KOUNT=KOUNT+1
      IF (KOUNT.LE.NFAR) GO TO 159
      IT=IT+1
      SUBIT=0
      FR=1
      DO 250 K=1,NFAR
      E(K)=.1
      D(K)=0.0
      A(K)=2.0
250   CONTINUE
      GO TO 11
1000  CALL EXIT
      END
      SUBROUTINE INTG(ALPHA,EJ)
      COMMON/ARRAY/XEMP(1000),SEMP(1000),EDD(1000),ELG(1000),TAU(1000)
     1,TX(1000),TS(1000)
      COMMON/S/STAL,C.,DEL,NSTP,ND,TS,IPT,Y10
      DIMENSION W(4),P(4,4),P1(4,4),P2(4,4),ALPHA(8),A(4,4)
     1,B(16),EB(16),EBINT(16),EA(4,4),EAINT(4,4),F(4),COC(4,4),D(4)
     2,X3(1000),X4(1000),EDH(1000),EDDH(1000)
      EQUIVALENCE (A(1,1),B(1)),(EA(1,1),EB(1)),(EAINT(1,1),EBINT(1))
      DATA WT/1./
C
C INITIALIZATION
C
      DO 1 I=1,ND
      DO 1 J=1,ND
      P(I,J)=0.
      COC(I,J)=0.
1     A(I,J)=0.
      NDIV=NSTP/3
      ND1=ND-1
      ND2=ND-2
      NNA=ND/2-1
      DO 11 I=1,ND
      W(I)=0.
      D(I)=0.
      F(I)=0.
11    P(I,I)=1.25 C
      P(1,1)=1.06 19937
      X3(1)=-1.06202
      X4(1)=0.
      EDH(1)=X3(1
      EDDH(1)=0.
      NDIM=ND
      N1=ND**2
      CB=COS(ELG(125))
      W(2)=-1.013351*CB
      W(1)=Y10*CB
      P(1,1)=P(1,1)*CB**2
      P(2,2)=(0.010626*CB)**2
      SMEAN=(W(1)/CB-XEMP(1))**2+(W(2)/CB-TX(1))**2
      SSD=(SQRT(P(1,1))/CB-SEMP(1))**2+(SQRT(P(2,2))/CB-TS(1))**2
```

```
      S1=0.
      DO 10 KK=1,NSIP
      K1=KK+1
      CB=COS(ELG(KK))
      ARG=-DEL*ALPHA(1)*CB
      IF(ARG.GT.-200.) S1=EXP(ARG)
      X3(K1)=X3(KK)+EDG(KK)*DEL
      IF(ALPHA(1).EQ.0.) GO TO 4
      X4(K1)=S1*(X4(KK)+EDD(KK)*DEL)
      GO TO 3
4     X4(K1)=X4(KK)+EDG(KK)*DEL
3     CONTINUE
      EDH(K1)=X3(K1)-X4(K1)
      IF(KK.GE.2) EDDH(KK)=(EDH(KK)-EDH(KK-1))/DEL
10    CONTINUE
      T=T0
      DO 100 KK=1,NSTP
      K1=KK+1
      M=TAU(KK)/DEL
      RA=NAA/TAU(KK)
      DO 2 I=1,ND2
      J1=I+2
      A(J1,I)=RA
      A(J1,J1)=-RA
2     CONTINUE
      THEDD=(ELG(K1)-ELG(KK))/DEL
      CB=COS(ELG(KK))
      TB=-THEDD*TAN(ELG(KK))
      SCAL=(G3*G3)**2/DEL
      T=T+DEL
      A2=1.-(TAU(K1)-TAU(KK))/DEL
      COR=CC*ALPHA(2)*CB
      A(1,1)=-COR+TB
      A(1,2)=-CC*ALPHA(3)*CB
      A(2,ND1)=-COR*A2
      A(2,ND)=A(1,2)*A2
      A(2,2)=TB
C
C-COMPUTE TRANSITION MATRIX EA AND ITS INTEGRAL EAINT
C
      CALL DSCRT(NDIM,E,DEL,EB,EBINT,5)
      CR4=CC*ALPHA(4)*CB
C
C START INTEGRATION LOOP
C
C     IF((MOD(KK, 5).EQ.0).OR.(KK.EQ.1)) PRINT 99,T,W(1),SMEAN,
C    1 SQRT(P(1,1)),SSD
99    FORMAT(5G12.5)
C
C COMPUTE MEAN TRACKING ERROR
C
      DO 110 I=1,NDIM
      DO 120 J=1,NDIM
      D(I)=D(I)+E (I,J)*W(J)
```

```
120   CONTINUE
110   CONTINUE
      F(1)=(CB-GR4)*X3(KK)+GR4*X4(KK)
      F(2)=X3(KK)*CB
      IF(KK.GT.M) F(2)=F(2)+GR4*(X4(KK-M)-X3(KK-M))*A2
      DO 130 I=1,NDIM
      DO 140 J=1,2
      D(I)=D(I)+EAINT(I,J)*F(J)
140   CONTINUE
      W(I)=D(I)
      D(I)=0.
130   CONTINUE
C
C COMPUTE ERROR DUE TO MEAN TRACKING ERROR
C
      SMEAN=SMEAN+(W(1)/CB-XEMP(K1))**2+(W(2)/CB-TX(K1-125))**2
C
C COMPUTE COVARIANCE MATRIX
C
      CQQ(1,1)=(ALPHA(5)+ALPHA(6)*ABS(EDH(KK))+ALPHA(7)*ABS(EDDH(KK)))
     1*SCAL
      CQQ(2,2)=ALPHA(5)*SCAL*A2**2
      IF(KK.GT.M) CQQ(2,2)=(ALPHA(5)+ALPHA(6)*ABS(EDH(KK-M))+ALPHA(7)*
     1 ABS(EDDH(KK-M)))*SCAL*A2**2
      CALL MULT(EAINT,CQQ,NDIM,N1,P1,10)
      CALL MULT(EA,P,NDIM,N1,P2,11)
      DO 220 I=1,NDIM
      DO 220 J=1,NDIM
      P(I,J)=P1(I,J)+P2(I,J)
220   CONTINUE
C
C COMPUTE ERROR DUE TO STANDARD DEVIATION
C
      SSD=SSD+(SQRT(P(1,1))/CB-SEMP(K1))**2+(SQRT(P(2,2))/CB-TS(K1-125))
     A**2
100   CONTINUE
      EJ=DELA*(SMEAN+WT*SSD)
      RETURN
      END
      SUBROUTINE MULT(E,F,L,L1,H,MR)
      DIMENSION E(1),F(1),G(16),H(1)
      DO 10 I=1,L
      II=1
      DO 10 K=1,L
      TEMP=0.
      DO 5 J=I,L1,L
      TEMP=TEMP+E(J)*F(II)
5     II=II+1
      KK=(K-1)*L+I
      H(KK)=TEMP
10    G(KK)=TEMP
      IF(MR.EQ.1)RETURN
      DO 20 I=1,L
      DO 20 K=I,L
      TEMP=0.
      II=K
      DO 15 J=I,L1,L
      TEMP=TEMP+G(J)*E(II)
15    II=II+L
      KK=(K-1)*L+I
```

```
20      H(KK)=TEMP
        L2=L-1
        DO 30 I=1,L2
        L3=I+1
        DO 30 J=L3,L
        K1=(I-1)*L+
        K2=(J-1)*L+1
30      H(K1)=H(K2)
        END
        SUBROUTINE ESCRT(NDIM,A,DEL,EA,EAINT,NT)
        DIMENSION A(1),EA(1),EAINT(1),COEF(30)
C          SETS EA=EXP(A*DEL),EAINT-INTEGRAL EA 0 TO DEL
        NDIM1=NDIM+1
        NN=NDIM*NDIM
        NTM1=NT-1
        COEF(NT)=1.
        DO 10 I=1,NTM1
        II=NT-I
   10   COEF(II)=DEL*COEF(II+1)/FLOAT(I)
C             NT MUST BE AT LEAST 3
        CALL DIAG(NDIM,EAINT,A,COEF(1),COEF(2))
        DO 60 L=3,NT
        CALL MULT(A,EAINT,NDIM,NN,EA,1)
        IF(L.EQ.NT)GO TO 70
   60   CALL DIAG(NDIM,EAINT,EA,1.0,COEF(L))
   70   DO 80 II=1,NN,NDIM1
        EA(II)=EA(II)+1.0
   80   CONTINUE
        END
        SUBROUTINE DIAG(NDIM,A,B,C1,C2)
        DIMENSION A(1),B(1)
        NDIM1=NDIM+1
        NN=NDIM*NDIM
        NM1=NDIM-1
        II=1
        IF(C1.EQ.1.0) GO TO 10
        DO 5 J=1,NN,NDIM
        K=J+NM1
        DO 4 I=J,K
    4   A(I)=C1*B(I)
        A(II)=A(II)+C2
    5   II=II+NDIM1
        RETURN
   10   DO 7 J=1,NN,NDIM
        K=J+NM1
        DO 6 I=J,K
    6   A(I)=B(I)
        A(II)=A(II)+C2
    7   II=II+NDIM1
        RETURN
        END
2000,4,30.0,2,0.026954
1.,1.,1.,1.,.0001,.0001,.0001
0.01
```

*.... ...   WARFICO //// END OF LIST ////

APPENDIX E

LISTING OF COMPUTER SIMULATION PROGRAM FOR

AN AAA TRACKING AND FIRING TASK

```
HOW,T20,CM70000,L760295,HEL,2563966
COMMENT.  SIMU6,ID=L760295,CY=1:AAA MODE6 SIMULATION PROGRAM
ATTACH,TAPE1,DATMERAZ,ID=L760295,CY=3,MR=1.
FTN.
LGO.
      PROGRAM ELFIT(INPUT,OUTPUT,TAPE1)
      COMMON/S/CO(2),DEL,IH,NDIM,Y10(2),X3(2),EL,ELDD,AZDD,NTAU,NA,A2,ND
     A1,ND2,NAA,UEL,UAZ,ELTR,AZTR,ISET,Z1,Z2
C
C THE PURPOSE OF THIS PROGRAM IS TO SIMULATE AN ELEVATION TRACKING
C TASK IN THE TRACER-DIRECTED-FIRE (MODE 6) SYSTEM
C INPUT: THE ELEVATION (EL) & AZIMUTH (AZ) ANGULAR ACCELERATION OF
C TARGET
C OUTPUT: MEAN AND STAND DEV OF LAG ANGLE
C ***** ALL ANGLES ARE IN UNITS OF RADIAN *****
C TAU: DELAY IN SECONDS
C ALPHA: PARAMETER VECTOR
C ELERR: MEAN EL LAG ANGLE (I.E. TARGET ANGLE-BARREL ANGLE)
C AZERR: MEAN AZ LAG ANGLE
C ELSD: STANDARD DEVIATION OF ELEVATION LAG ANGLE
C AZSD: STANDARD DEVIATION OF AZ LAG ANGLE
C ELTR: MEAN EL TRACER ERROR (TARGET ANGLE-TRACER ENDING ANGLE)
C AZTR: MEAN AZ TRACER ERROR
C ELBAR: MEAN EL  ARREL ANGLE
C DEL: TIME STEP USED IN THE INTEGRATION ROUTINE
C Y10(1): INITIAL GUESS OF EL LAG ANGLE
C Y10(2): INITIAL GUESS OF AZ LAG ANGLE
C UEL: EL CONTROL
C UAZ: AZ CONTROL
C CO(1): EL RATE CONTROL COEFF
C CO(2): AZ RATE CONTROL COEFF
C K1: NO OF POINTS IN THE ENTIRE TRAJECTORY
C K: NO OF POINTS AFTER THE FIRST TRACER ROUND IS FIRED
C ELDD: EL ANGULAR ACCELERATION OF TARGET
C AZDD: AZ ANGULAR ACCELERATION OF TARGET
C X3(1): EL ANGULAR VELOCITY OF TARGET
C X3(2): AZ ANGULAR VELOCITY OF TARGET
C X4: ESTIMATION ERROR OF ANGULAR VELOCITY OF TARGET
C EL: EL ANGULAR POSITION OF TARGET
C W(1): MODEL PREDICTED LAG ANGLE
C W(2): MODEL PREDICTED TRACER ERROR
C P(1,1): VARIANCE OF PREDICTED LAG ANGLE
C P(2,2): VARIANCE OF PREDICTED TRACER ERROR
C T0: THE INITIAL FIRING TIME
C
      READ*,K1,T0,IPT
      PRINT 3,K1,T0,IPT
3     FORMAT(1H1,"NO OF PTS= ",I4,2X,"INIT TIME= ",G12.5//1X,"READ EVERY
     C",I2," POINT"/)
      KT=T0/0.03
      K=K1-KT
      T=T0
      NDIM=4
      IPRINT=20/IPT
      CO(1)=1.34
      CO(2)=1.24
      NE1=NDIM-1
      NE2=NDIM-2
```

```
      NAA=NDIM/2-1
      OTAU=7.5
      DEL=0.03*IPT
      ELSD=.0002* 0.5
      AZSD=.0002* 0.5
      Z1=0.
      Z2=0.
      ISET=0
      UEL=0.
      UAZ=0.
      PRINT 7
7     FORMAT(1H ,2X,"TIME",8X,"EL VEL",9X,"ELERR ",6X,"ELSD",6X,
     1"EL GTR",6X,"AZ VEL",6X,"AZERR",6X,"AZSD",6X,"AZ GTR"
     2,6X,"EL TR",6X,"AZ TR"/)
      READ(1,2)(DUM1,AZ,AZD,AZDD,EL,ELD,ELDD,T,AZMN,X,ZSD,S,I=1,KT
     C )
      DO 5 I=1,K
      READ(1,2)DUM1,AZ,AZD,AZDD,EL,ELD,ELDD,T,AZMN,X,ZSD,S
      IF(EOF(1))1,1
1     IF(MOD(I-1,IPT).NE.0) GO TO 5
      IH=(I-1)/IPT+1
      T=TC+(IH-1) DEL
      TAU=7.5
      IF(DUM1.LE.2877.) TAU=DUM1/(930.-.19*DUM1)
      MTAU=TAU/DEL
      KA=NAA/TAU
      A2=1.-(TAU-OTAU)/DEL
      OTAU=TAU
      X3(1)=ELD
      X3(2)=AZD
      IF((IH-1).GE.MTAU) GO TO 9
      TAUR=TAU
      IF(DUM1.LE.4450) TAUR=TAU*AMAX1(1.6,DUM1/5030.)
      Y10(1)=-TAUR*ELD-.001*(5.2*TAUR+.486*TAUR**2)*COS(EL+0.05)
      Y102=-3.025 SIGN(1.,AZD)
      IF(IH.NE.1) GO TO 10
      EL0=EL
      AZ0=AZ
      E10=Y10(1)
      E20=Y102
      GO TO 11
9     ISET=ISET+1
      IF(ISET.NE.1) GO TO 10
      Z1=EL-(EL0-E10)+.001*(5.2*TAU+.486*TAU**2)*COS(ELC+0.05)
      Z2=AZ-(AZ0-E20)
10    CALL OBSEL6(ELERR,ELSD)
      ELBAR=EL-ELERR
      IF((IH-1).LT.MTAU) Y10(2)=Y102*COS(ELBAR)
      CALL OBSAZ6(AZERR,AZSD,ELBAR)
      IF((MOD(IH-1,IPRINT).EQ.0).OR.(IH.EQ.1)) PRINT 6,T,X3(1),ELERR
     1,ELSD,UEL,X3(2),AZERR,AZSD,UAZ,ELTR,AZTR
6     FORMAT(11G12.5)
5     CONTINUE
2     FORMAT(12G11.5)
      STOP
      END
```

```
      SUBROUTINE CBSEL6(ELERR,ELSD)
      COMMON/S/CG(2),DEL,KK,ND,Y10(2),X30(2),EL,EDD,AZDD,M,RA,A2
     A,ND1,ND2,NAA,U,UAZ,ELTR,AZTR,ISET,Z1,Z2
      DIMENSION W(4),F(4,4),P1(4,4),P2(4,4),ALPHA(8),A(4,4)
     1,B(16),EB(16),EBINT(16),EA(4,4),EAINT(4,4),F(4),COC(4,4),D(4)
     2,X3(1000),X4(1000),EDH(1000),EDDH(1000)
      EQUIVALENCE (A(1,1),B(1)),(EA(1,1),EB(1)),(EAINT(1,1),EBINT(1))
      DATA WT/1./
      DATA ALPHA/1.2862,.79531,1.,1.43,1.4932,.90605,.1E-4,.22807E-5,
     A.75607E-3/
C
C INITIALIZATION
C
      IF((KK-1).GE.M) GO TO 6
      IF(KK.GT.1) GO TO 5
      N1=ND**2
      SCAL=CG(1)**2/DEL
      DO 1 I=1,ND
      DO 1 J=1,ND
      P(I,J)=0.
      COC(I,J)=0.
1     A(I,J)=0.
      DO 11 I=1,ND
      W(I)=0.
      D(I)=0.
      F(I)=0.
11    P(I,I)=0.0D 0
      P(1,1)=0.0D 2
      P(2,2)=0.0D 2
      X3(1)=X30(1)
      X4(1)=0.
      EDH(1)=X3(1)
      EDDH(1)=0.
      S1=0.
      ARG=-DEL*ALPHA(2)
      IF(ARG.GT.-200.) S1=EXP(ARG)
      NDIM=ND
      COR=CG(1) ALPHA(3)
      CR4=CG(1) ALPHA(5)
5     W(1)=Y10(1)
6     IF(ISET.EQ.1) W(2)=Z1
      K1=KK+1
      K2=KK-1
C
C COMPUTE TARGET VELOCITY AND ESTIMATION ERROR
C
      X3(K1)=X3(KK)+EDD*DEL
      IF(ALPHA(2).EQ.0.) GO TO 4
      X4(K1)=S1*X4(KK)+EDD*(1.-S1)/ALPHA(2)
      GO TO 3
4     X4(K1)=X4(KK)+EDD*DEL
3     CONTINUE
      EDH(KK)=X3(KK)-X4(KK)
      IF(KK.GE.2) EDDH(KK)=(EDH(KK)-EDH(K2))/DEL
      X30(1)=X3(K1)
      IF((KK-1).LE.M) GO TO 150
      A(1,1)=-COR
```

```
      A(1,2)=-CJ(1) ALPHA(4)
      A(2,NC-1)=A(1,1)*ALPHA(1)*A2
      A(2,NC)=A(1 2)*ALPHA(1)*A2
      DO 2 I=1,NJ2
      J1=I+2
      A(J1,I)=RA
      A(J1,J1)=-R
2     CONTINUE
      CALL ESCRT(NC,G,CEL,EB,EBINT,5)
      CR3=CR+ALPHA(1)*A2
      SCAL1=SCAL*(ALPHA(1)*A2)**2
      CGC(2,2)=ALPHA(6)*SCAL1
C
C COMPUTE MEAN TRACKING ERROR (I.E. LAG ANGLE)
C
      U=ALPHA(3)*W(1)+ALPHA(4)*W(2)+ALPHA(5)*(X3(KK)-X4(KK))
      DO 110 I=1,NC
      DO 120 J=1,NC
      D(I)=D(I)+EA(I,J)*W(J)
120   CONTINUE
110   CONTINUE
      F(1)=(1.-CR4)*X3(KK)+CR4*X4(KK)
      F(2)=X3(KK)+CR3*(X4(KK-M)-X3(KK-M))
      DO 130 I=1,NC
      DO 140 J=1,2
      D(I)=D(I)+EAINT(I,J)*F(J)
140   CONTINUE
      W(I)=D(I)
      D(I)=0.
130   CONTINUE
C
C COMPUTE COVARIANCE MATRIX
C
      CGC(1,1)=(ALPHA(6)+ALPHA(7)*ABS(EDH(KK))+ALPHA(8)*ABS(EDDH(KK)))
     1 *SCAL
      CGC(2,2)=(ALPHA(6)+ALPHA(7)*ABS(EDH(KK-M))+ALPHA(8)*ABS(EDDH(KK-M)
     1))*SCAL1
      CALL MULT(EAINT,CGC,NC,N1,P1,10)
      CALL MULT(EA,F,NC,N1,P2,10)
      DO 220 I=1,NC
      DO 220 J=1,NC
      P(I,J)=P1(I,J)+P2(I,J)
220   CONTINUE
150   CONTINUE
      ELERR=W(1)
      ELSD=SQRT(P(1,1))
      ELTR=W(2)
      RETURN
      END
      SUBROUTINE GESAZ6(AZERR,AZSD,ELG)
      COMMON/3/CO(2),DEL,KK,ND,Y10(2),X30(2),EL,ELDD,AZDD,M,RA,
     1 A2,NC1,NC2,NAA,UEL,U,ELTR,AZTR,ISET,Z1,Z2
      DIMENSION W(4),P(4,4),P1(4,4),P2(4,4),ALPHA(8),A(4,4)
     1,B(16),EB(16),EBINT(16),EA(4,4),EAINT(4,4),F(4),CGC(4,4),C(4)
     2,X3(1000),X4(1000),EDH(1000),EDDH(1000)
      EQUIVALENCE (A(1,1),B(1)),(EA(1,1),EB(1)),(EAINT(1,1),EBINT(1))
      DATA WT/1./
      DATA ALPHA/557.15,.15447,.61586,.93641,.15634E-4,.54182E-4,.005289
     A/
```

```
      IF((KK-1).GE.M) GO TO 6
      IF(KK.GT.1) GO TO 5
      N1=ND**2
C
C INITIALIZATION
C
      DO 1 I=1,ND
      DO 1 J=1,ND
      P(I,J)=0.
      CGC(I,J)=0.
1     A(I,J)=0.
      DO 11 I=1,ND
      W(I)=0.
      U(I)=0.
      F(I)=0.
11    P(I,I)=1.0D00
      P(1,1)=1.0D02 COS(ELG)**2
      P(2,2)=1.0D02 COS(ELG)**2
      X3(1)=X30(2)
      X4(1)=0.
      EDH(1)=X3(1)
      EDDH(1)=0.
      S1=0.
C
C COMPUTE AND STORE STATES X3 AND X4
C
5     W(1)=Y1J(2)
6     IF(ISET.EQ.1) W(2)=Z2*COS(ELG)
      CONTINUE
      CB=COS(ELG)
      ARG=-DEL*ALPHA(1)*CB
      IF(ARG.GT.-200.) S1=EXP(ARG)
      K1=KK+1
      K2=KK-1
      X3(K1)=X3(KK)+AZDD*DEL
      IF(ALPHA(1).EQ.0.) GO TO 4
      X4(K1)=S1*(X4(KK)+AZDD*DEL)
      GO TO 3
4     X4(K1)=X4(KK)+AZDD*DEL
3     CONTINUE
C
C COMPUTE AND STORE ESTIMATED TARGET VELOCITY AND ACCELERATION
C
      EDH(K1)=X3(K1)-X4(K1)
      IF(K2.GE.1) EDDH(KK)=(EDH(KK)-EDH(K2))/DEL
      X30(2)=X3(K1)
      IF((KK-1).LE.M) GO TO 150
      DO 2 I=1,ND2
      J1=I+2
      A(J1,I)=RA
      A(J1,J1)=-RA

      THEDD=(ELG-OELG)/DEL
      CB=COS(ELG)
      TB=-THEDD*TAN(ELG)
      SCAL=(CD(2) CL)**2/DEL
      CGR=CC(2)-ALPHA(2)*CB
```

```
      A(1,1)=-CFR+TR
      A(1,2)=-CU(2)-ALPHA(3)*CB
      A(2,ND1)=-CJR-A2
      A(2,N( )=A(1,2)*A2
      A(2,2)=TB
C
C COMPUTE TRANSITION MATRIX EA AND ITS INTEGRAL EAINT
C
      CALL DSCRT(NC,J,DEL,EB,EBINT,5)
      CR4=GC(2)-ALPHA(4)*CB
      CQC(2,2)=ALPHA(5)+SCAL*A2**2
C
C COMPUTE MEAN TRACKING ERROR
C
      U=ALPHA(2)*W(1)+ALPHA(3)*W(2)+ALPHA(4)*(X3(KK)-X4(KK))
      DO 110 I=1,NC
      DO 120 J=1,NC
      D(I)=D(I)+EA(I,J)*W(J)
120   CONTINUE
110   CONTINUE
      F(1)=(C3-CR4) X3(KK)+CR4*X4(KK)
      F(2)=X3(KK)-C +CR4*(X4(KK-M)-X3(KK-M))*A2
      DO 130 I=1,NC
      DO 140 J=1,2
      D(I)=D(I)+EAINT(I,J)*F(J)
140   CONTINUE
      W(I)=D(I)
      D(I)=0.
130   CONTINUE
C
C COMPUTE COVARIANCE MATRIX
C
      CQC(1,1)=(ALPHA(5)+ALPHA(6) ABS(EDH(KK))+ALPHA(7)*ABS(EDDH(KK)))
     1 *SCAL
      CQC(2,2)=(ALPHA(5)+ALPHA(6) ABS(EDH(KK-M))
     1 +ALPHA(7)*ABS(EDDH(KK-M)))*SCAL*A2**2
      CALL MULT(EAINT,CQC,ND,N1,P1,10)
      CALL MULT(EA,P,ND,N1,P2,10)
      DO 220 I=1,NC
      DO 220 J=1,NC
      P(I,J)=P1(I,J)+P2(I,J)
220   CONTINUE
150   CONTINUE
      DELG=ELG
      AZERR=W(1)/CB
      AZSD=SQRT(P(1,1))/CB
      AZTR=W(2)/CB
      RETURN
      END
      SUBROUTINE MULT(E,F,L,L1,H,MR)
      DIMENSION E(1),F(1),G(16),H(1)
      DO 1  I=1,L
      II=1
      DO 10 K=1,L
      TEMP=0.
      DO 5 J=I,L1,L
      TEMP=TEMP+E(J)*F(II)
```

```
5     II=II+1
      KK=(K-1)*L+I
      H(KK)=TEMP
10    G(KK)=TEMP
      IF(MR.EQ.1)RETURN
      DO 20 I=1,L
      DO 20 K=I,L
      TEMP=0.
      II=K
      DO 15 J=I,L1,L
      TEMP=TEMP+G(J)*E(II)
15    II=II+L
      KK=(K-1)*L+I
20    H(KK)=TEMP
      L2=L-1
      DO 30 I=1,L2
      L3=I+1
      DO 30 J=L3,L
      K1=(I-1)*L+J
      K2=(J-1)*L+I
30    H(K1)=H(K2)
      END
      SUBROUTINE ESERT(NDIM,A,DEL,EA,EAINT,NT)
      DIMENSION A(1),EA(1),EAINT(1),COEF(30)
C        SETS EA=EXP(A*DEL),EAINT=INTEGRAL EA 0 TO DEL
      NDIM1=NDIM+1
      NN=NDIM*NDIM
      NTM1=NT-1
      COEF(NT)=1.
      DO 10 I=1,NTM1
      II=NT-I
   10 COEF(II)=DEL*COEF(II+1)/FLOAT(I)
C           NT MUST BE AT LEAST 3
      CALL DIAG(NDIM,EAINT,A,COEF(1),COEF(2))
      DO 60 L=3,NT
      CALL MULT(A,EAINT,NDIM,NN,EA,1)
      IF(L.EQ.NT)GO TO 70
   60 CALL DIAG(NDIM,EAINT,EA,1.0,COEF(L))
   70 DO 80 II=1,NN,NDIM1
      EA(II)=EA(II)+1.0
   80 CONTINUE
      END
      SUBROUTINE DIAG(NDIM,A,B,C1,C2)
      DIMENSION A(1),B(1)
      NDIM1=NDIM+1
      NN=NDIM*NDIM
      NM1=NDIM-1
      II=1
      IF(C1.EQ.1. ) GO TO 10
      DO 5 J=1,NN,NDIM
      K=J+NM1
      DO 4 I=J,K
    4 A(I)=C1*B(I)
      A(II)=A(II)+C2
    5 II=II+NDIM1
      RETURN
   10 DO 7 J=1,NN,NDIM
      K=J+NM1
      DO 6 I=J,K
    6 A(I)=B(I)
      A(II)=A(II)+C2
    7 II=II+NDIM1
      RETURN
      END
2208,30.6,2
```

REFERENCES

Banks, H. T. and J. A. Burns, March 1978, "Hereditary Control Problems: Numerical Methods Based on Average Approximations," SIAM J. Control and Optimization, Vol. 16, No. 2, pp. 169-208.

Kleinman, D. L., S. Baron, and W. H. Levison, December 1971, "A Control Theoretic Approach to Manned-Vehicle Systems Analysis," IEEE Trans. Automat. Contr., Vol. AC-16, No. 6, pp. 824-832.

Kleinman, D. L. and T. R. Perkins, August 1974, "Modeling Human Performance in a Time-Varying Anti-Aircraft Tracking Loop," IEEE Trans. Automat. Contr., Vol. AC-19, No. 4, pp. 297-306.

Kou, R. S. and B. C. Glass, August 1979, "Development of Observer Model for AAA Tracker Response," AMRL-TR-79-77, (AD A080973), Aerospace Medical Research Laboratory, Wright-Patterson AFB, Ohio.

Kou, R. S., et al., 1978, "A Linear Time-Varying Antiaircraft Gunner Model Based on Observer Theory," Proceedings of the International Conference on Cybernetics and Society, Vol. II, pp. 1421-1426.

Luenberger, D. G., December 1971, "An Introduction to Observers," IEEE Trans. Automat. Contr., Vol. AC-16, No. 6, pp. 596-602.

McRuer, D. T. and E. S. Krendel, 1974, "Mathematical Models of Human Pilot Behavior," AGARD-AG-188.

Phatak, A., H. Weinert, I. Segall, and C. N. Day, 1976, "Identification of a Modified Optimal Control Model for the Human Operator," Automatica, Vol. 12, pp. 31-41.

Rolek, E. P., September 1977, "MTQ Experimentation Report No. 5--MTQ AZ EL One Study Initial Analysis," Systems Research Laboratories, Inc., Report No. 6874-15-18, pp. 13-14.

Rosenbrock, H., ed., 1966, Computational Techniques for Chemical Engineers, Pergamon Press, pp. 64-68.

☆U.S.Government Printing Office: 1981 — 757-002/448